



DUDLEY K. JOE LIBRARY  
NAVAL POSTGRADUATE SCHOOL  
MORTENBY, CALIFORNIA 93945-2002





# NAVAL POSTGRADUATE SCHOOL

## Monterey, California



# THESIS

CMOS CELL LIBRARY FOR A SILICON COMPILER

by

Anthony Joseph Mullarky

March 1987

Thesis Advisor:

D. E. Kirk

Approved for public release; distribution is unlimited

T233308





## REPORT DOCUMENTATION PAGE

|  |       |  |   |  |                           |
|--|-------|--|---|--|---------------------------|
| 1a REPORT SECURITY CLASSIFICATION<br>UNCLASSIFIED  |       |  | 1b RESTRICTIVE MARKINGS   |  |                           |
| 2a SECURITY CLASSIFICATION AUTHORITY   |       |  | 3 DISTRIBUTION/AVAILABILITY OF REPORT<br>Approved for public release;<br>distribution is unlimited. |  |                           |
| 2b DECLASSIFICATION/DOWNGRADING SCHEDULE   |       |  | 5 MONITORING ORGANIZATION REPORT NUMBER(S)  |  |                           |
| 4 PERFORMING ORGANIZATION REPORT NUMBER(S)   |       |  | 7a NAME OF MONITORING ORGANIZATION<br>Naval Postgraduate School                                     |  |                           |
| 6a NAME OF PERFORMING ORGANIZATION<br>Naval Postgraduate School  |       | 6b OFFICE SYMBOL<br>(if applicable)<br>62      | 7b ADDRESS (City, State, and ZIP Code)<br>Monterey, California 93943-5000                           |  |                           |
| 6c ADDRESS (City, State, and ZIP Code)<br>Monterey, California 93943-5000  |       | 9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER |   |  |                           |
| 8a NAME OF FUNDING/SPONSORING ORGANIZATION   |       | 8b OFFICE SYMBOL<br>(if applicable)            | 10 SOURCE OF FUNDING NUMBERS  |  |                           |
| 8c ADDRESS (City, State, and ZIP Code)   |       | PROGRAM ELEMENT NO                             |   |  |                           |
|  |       | PROJECT NO                                     |   |  |                           |
|  |       | TASK NO  |   |  |                           |
|  |       | WORK UNIT<br>ACCESSION NO                      |   |  |                           |
| 11 TITLE (Include Security Classification)<br>CMOS CELL LIBRARY FOR A SILICON COMPILER   |       |  |   |  |                           |
| 12 PERSONAL AUTHOR(S)<br>Mullarky, Anthony J.  |       |  |   |  |                           |
| 13a NAME OF REPORT<br>Master's Thesis  |       | 13b TIME COVERED<br>FROM TO                    |   | 14 DATE OF REPORT (Year Month Day)<br>1987 March |                           |
|  |       |  |   | 15 PAGE COUNT<br>108                             |                           |
| 16 SUPPLEMENTARY NOTATION  |       |  |   |  |                           |
| 17 COSATI CODES  |       |  | 18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)                    |  |                           |
| FIELD  | GROUP | SUB-GROUP                                      | CMOS, CMOS Cell Library, CMOS Organelles,<br>CMOS Silicon Compiler Organelles                       |  |                           |
|  |       |  |   |  |                           |
| 19 ABSTRACT (Continue on reverse if necessary and identify by block number)<br>A standard Complementary Metal Oxide Silicon (CMOS) library for use in Very Large Scale Integration (VLSI) circuits was developed. The development includes investigation of the various clocking strategies upon which the optimum clocking strategy, pseudo-two phase, was selected for all clocked cells in the library. The cells were then designed using the pseudo-two phase clocking strategy. A primary objective is to provide cells for use in converting the MACPITTS silicon compiler from n-channel Metal Oxide Silicon (NMOS) to Cmos technology. Cell layouts, timing data, schematics and logic tables for each cell are provided. |       |  |   |  |                           |
| 20 DISTRIBUTION/AVAILABILITY OF ABSTRACT<br><input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS  |       |  | 21 ABSTRACT SECURITY CLASSIFICATION<br>Unclassified   |  |                           |
| 22a NAME OF RESPONSIBLE INDIVIDUAL<br>Kirk, D.E.   |       |  | 22b TELEPHONE (Include Area Code)<br>(408) 646-2056   |  | 22c OFFICE SYMBOL<br>62KI |

Approved for public release; distribution is unlimited.

**CMOS Cell Library for a  
Silicon Compiler**

by

Anthony Joseph Mullarky  
Lieutenant, United States Navy  
B.S.E.E., University of Florida, 1980

Submitted in partial fulfillment of the  
requirements for the degree of

**MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL**

March 1987



## ABSTRACT

A standard Complementary Metal Oxide Silicon (CMOS) library for use in Very Large Scale Integration (VLSI) circuits was developed. The development includes investigation of the various clocking strategies upon which the optimum clocking strategy, pseudo-two phase, was selected for all clocked cells in the library. The cells were then designed using the pseudo-two phase clocking strategy. A primary objective is to provide cells for use in converting the MACPITTS silicon compiler from n-channel Metal Oxide Silicon (NMOS) to CMOS technology. Cell layouts, timing data, schematics and logic tables for each cell are provided.

Thesis  
M891  
C-1

## TABLE OF CONTENTS

|   |    |
|---|----|
| I. INTRODUCTION .....                             | 6  |
| A. BACKGROUND .....                               | 6  |
| B. GOALS .....                                    | 7  |
| 1. CMOS Versus NMOS .....                         | 7  |
| 2. Selecting Clocking Strategies for CMOS .....   | 8  |
| 3. Hierarchical Cells Versus Standard Cells ..... | 8  |
| C. IMPLEMENTATION .....                           | 10 |
| II. CLOCKING STRATEGIES .....                     | 11 |
| A. SINGLE-PHASE CLOCKING .....                    | 11 |
| B. TWO-PHASE CLOCKING .....                       | 17 |
| C. APPLICATIONS .....                             | 22 |
| D. CONCLUSIONS .....                              | 23 |
| III. LAYOUT PHILOSOPHY .....                      | 24 |
| A. SCHEMATIC GENERATION .....                     | 24 |
| B. SPICE SIMULATIONS .....                        | 25 |
| 1. Sizing Transistors .....                       | 26 |
| 2. Circuit Functionality .....                    | 27 |
| 3. Propagation Delays .....                       | 27 |

|   |     |
|---|-----|
| C. STICK DIAGRAMS .....                 | 28  |
| D. MAGIC USAGE FOR STANDARD CELLS ..... | 28  |
| E. CHECKING LAYOUTS .....               | 30  |
| IV. APPLICATIONS .....                  | 32  |
| A. CIRCUIT SIZE COMPARISION .....       | 32  |
| B. SIMULATION RESULTS .....             | 33  |
| C. APPLICATION EXAMPLES .....           | 34  |
| V. CONCLUSIONS .....                    | 37  |
| APPENDIX ORGANELLES .....               | 39  |
| LIST OF REFERENCES .....                | 104 |
| INITIAL DISTRIBUTION LIST .....         | 105 |

## I. INTRODUCTION

### A. BACKGROUND

A silicon compiler is an automatic translation tool that takes a behavioral description written in a high level language, such as LISP, and converts it to a mask level layout. The majority of silicon compilers are technology driven. That is, as new technologies are developed the research in silicon compilers is driven towards that technology. This leaves previously developed compilers, such as the University of Edinburgh's FIRST compiler [Ref. 1:p. 33] or the MACPITTS compiler [Ref. 2:pp. 2-5], obsolete every time a new technology is generated.

A better approach is to make a compiler technology independent, such as in the GENESIL compiler [Ref. 3:pp. 52-53]. This way when a new technology is developed all that needs to be added to the compiler are the new design rules and organelles for that technology. Since most compilers are characterized by a fixed floor plan this should be an easy task.

The MACPITTS silicon compiler uses an n-channel Metal Oxide Silicon (NMOS) database for its organelles (bit slice of an operator or register). Since it has a fixed floor plan, adding technologies should be straightforward. To demonstrate the possibility of doing this, the thesis project described herein is concerned with the design of a standard set of Complementary Metal Oxide Silicon (CMOS) organelles for insertion into the MACPITTS silicon compiler.

## B. GOALS

This thesis investigates the design of an expandable technology library for MACPITTS. The project is motivated by the shift in industry from NMOS to CMOS. To demonstrate the feasibility, a standard set of CMOS organelles (Appendix) was generated and inserted into MACPITTS. By designing the organelles to be functionally the same as their NMOS counterparts, the new cells will be able to use the existing MACPITTS test structures.

The resulting dual technology silicon compiler also incorporates a more efficient clocking strategy. Since the NMOS version of MACPITTS is implemented with a three-phase clock (much more conservative than necessary) the CMOS version attempts to use a more efficient two-phase clocking scheme.

### 1. CMOS Versus NMOS

Although any technology could have been used to examine the idea of an expandable technology library, CMOS was selected for several reasons. First, with a shift in industry from NMOS to CMOS the latter seems like an appropriate choice. Secondly, the two technologies are compatible in many ways [Ref. 4:pp. 1-28].

The major advantages of using CMOS over NMOS are the symmetry of CMOS which encourages symmetrical layout styles, the equal rise and fall times of CMOS transitions and lower power consumption. These advantages benefit circuit design in CMOS. The regular layout styles allow for easy determination of



transistor sizes. Because of equal rise and fall times, critical paths have the same propagation delays for rising and falling transitions.

A disadvantage of static CMOS is the number of transistors required. CMOS requires  $2N$  transistors for static complementary gates while NMOS only requires  $N+1$  transistors for  $N$  inputs. Thus, CMOS requires more chip area than NMOS. A more detailed analysis of CMOS versus NMOS is presented in [Ref. 4:pp. 1-28].

## 2. Selecting Clocking Strategies for CMOS

Various methods of clocking CMOS circuits to be used in MACPITTS were investigated. To augment the fragmentary information in the literature much of the necessary data was generated using computer models. Currently MACPITTS uses a conservative three-phase clocking scheme [Ref. 2:pp. 12-13]. Since a goal of this thesis investigation is to use a more efficient clocking scheme, three and four-phase clocking schemes are not considered because they increase circuit complexity and area without a significant gain in prevention of races caused by clock skew.

## 3. Hierarchical Cells Versus Standard Cells

MACPITTS NMOS organelles use a hierarchical layout style, that is, the building blocks consists of pull-up transistors, input structures, output structures, etc. The building blocks are assembled to build bigger building blocks, such as inverters, which in turn are assembled to form organelles. This slows down the execution of MACPITTS because every time an organelle is generated its building

blocks must be called, and in turn each of these must call up their building blocks. It is this sequential calling that increases compilation time.

There are two advantages of using hierarchical cells. First, hierarchical cells result in quicker hand generated layouts and are easier to check for design errors since the cells are constructed of pre-checked blocks. Secondly, once a mistake is discovered only the building block in error needs to be corrected and all the organelles using that building block receive the correction.

There are also several major disadvantages of using hierarchical layouts. First, using building blocks results in larger layouts because this type of layout style does not take full advantage of chip area. Secondly, if a mistake occurs in a building block, all organelles that use the structure must be checked for design rule violations after the building block is corrected. This is especially true if the correction involves increasing the building block's size, and since this results in a larger layout, the organelle will have a higher propagation delay due to the added resistance and capacitance.

A simpler method is to use a standard cell layout style. This method results in a stand-alone organelle. All the building blocks are assembled in a fixed structure in the organelle, that is, there is no hierarchy in the organelle. The advantages of this method are that it results in smaller layouts, and thus smaller propagation delays, and only the one organelle needs to be checked if a change is made to its layout. Disadvantages of this type of layout style are that it takes longer to layout an organelle because of its relative complexity and it is more

difficult to check for design rule violations because all building blocks are at the same hierarchical level in the organelle. The disadvantages result from standard cell layouts containing all the building blocks which are checked upon layout completion. In contrast, hierarchical layouts use pre-checked building blocks so that upon layout completion all that needs to be checked is the placement of the building blocks. The benefits of a standard cell layout style outweigh those of a hierarchical layout style for silicon compilation. Thus, the standard cell layout style was used for the layout of all CMOS organelles.

### C. IMPLEMENTATION

The following three chapters cover selection of a clocking strategy, guidelines for organelle layouts, and applications to a CMOS implemented MACPITTS. MAGIC CAD tools [Ref. 5:pp. 143-246] and the SPICE simulation package [Ref. 6] were used extensively in this investigation. Wherever possible MAGIC and SPICE terminology will be used.

## II. CLOCKING STRATEGIES

### A. SINGLE-PHASE CLOCKING

The D latch shown in Figure 2.1 is a single-phase latch that operates well with a clock whose complement has no lag with respect to the true clock (Figure 2.2) [Ref. 4:pp. 175-225]. During the load cycle of the latch, when the clock goes low, transmission gate T1 turns on and transmission gate T2 turns off. This is the ideal situation where no lag exists between the clock and its complement. However, in a non-ideal situation where the clock's  $\bar{\Phi}$  phase lags the  $\Phi$  phase, the p-channel transistor in T1 turns on while the n-channel transistor remains off until the positive level of the clock's complement arrives. For transmission gate T2 just the reverse is true: when  $\Phi$  goes high the n-channel transistor turns on while the p-channel transistor remains off until  $\bar{\Phi}$  arrives. The lag causes unacceptable operating conditions. Because the n-channel transistor in T2 is on and the p-channel transistor in T1 is on for the time when  $\Phi$  is high and  $\bar{\Phi}$  lags, there exists a direct path from the output Q of the latch to its input D. Thus, a logical one on Q can cause a logical zero on D to change due to the feedback path. To eliminate the feedback requires eliminating the clock lag. This is virtually impossible to do. There will always be a lag associated with the clock due to the delay through the circuit that generates the clock's complement.

Even if the circuit delay was eliminated through clever circuit design, there would be a lag caused by the delay from unequal clock line lengths on the chip.

Circuit simulations of the D latch using SPICE verified the above findings. MOSIS transistor parameters (Table 2.1) were used with channel lengths of  $3.0\mu\text{m}$

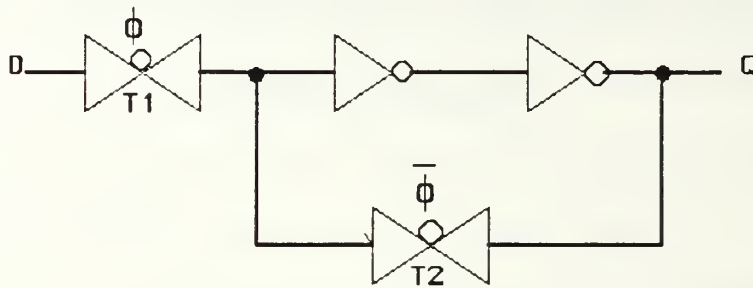


Figure 2.1 D Latch, Single Phase

---

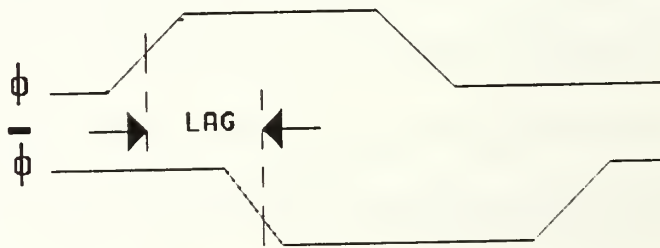


Figure 2.2 CMOS Single Phase Clock With Lag

---



and channel widths of  $4.5\mu\text{m}$  for all transistors in the circuit. A 5V supply and a 0.6ns delay through the inverter used to generate the complement of the clock resulted in a 0.63V feedback to D from Q. As the lag increases through greater delay in the inverter or through delays in unequal clock line lengths the feedback voltage also increases. A large enough lag can cause the feedback to increase to the point where D will change states. The feedback paths created by clock lag makes this circuit an unlikely candidate for MACPITTS.

TABLE 2.1 MOSIS TRANSISTOR PARAMETERS

| TYPE   | NOMINAL  |          | WORST CASE |          |
|--------|----------|----------|------------|----------|
|        | NMOS     | PMOS     | NMOS       | PMOS     |
| LEVEL  | 2.000    | 2.000    | 2.000      | 2.000    |
| VTO    | 0.827    | -0.895   | 0.909      | -0.984   |
| KP     | 3.29d-05 | 1.53d-05 | 3.29d-05   | 1.53d-05 |
| GAMMA  | 1.360    | 0.879    | 1.360      | 0.879    |
| PHI    | 0.600    | 0.600    | 0.600      | 0.600    |
| LAMBDA | 1.60d-02 | 4.71d-02 | 1.60d-02   | 4.71d-02 |
| CGSO   | 5.20d-10 | 4.00d-10 | 5.20d-10   | 4.00d-10 |
| CGDO   | 5.20d-10 | 4.00d-10 | 5.20d-10   | 4.00d-10 |
| RSH    | 25.000   | 95.000   | 25.000     | 95.000   |
| CJ     | 3.20d-04 | 2.00d-04 | 3.20d-04   | 2.00d-04 |
| MJ     | 0.500    | 0.500    | 0.500      | 0.500    |
| CJSW   | 9.00d-10 | 4.50d-10 | 9.00d-10   | 4.50d-10 |
| MJSW   | 0.330    | 0.330    | 0.330      | 0.330    |
| TOX    | 5.00d-08 | 5.00d-08 | 5.00d-08   | 5.00d-08 |
| NSUB   | 1.00d+16 | 1.12d+14 | 1.00d+16   | 1.12d+14 |
| NSS    | 0. d+00  | 0. d+00  | 0. d+00    | 0. d+00  |
| NFS    | 1.23d+12 | 8.79d+11 | 1.23d+12   | 8.79d+11 |
| TPG    | 1.000    | -1.000   | 1.000      | -1.000   |
| XJ     | 4.00d-07 | 4.00d-07 | 4.00d-07   | 4.00d-07 |
| LD     | 2.80d-07 | 2.80d-07 | 2.80d-07   | 2.80d-07 |
| UO     | 200.000  | 100.000  | 130.000    | 65.000   |
| UCRIT  | 9.99d+05 | 1.64d+04 | 9.99d+05   | 1.64d+04 |
| UEXP   | 0.001    | 0.153    | 0.001      | 0.153    |
| VMAX   | 1.00d+05 | 1.00d+05 | 1.00d+05   | 1.00d+05 |
| NEFF   | 0.010    | 0.010    | 0.010      | 0.010    |
| DELTA  | 1.241    | 1.938    | 1.214      | 1.938    |
| TEMP   | 27.00C   | 27.00C   | 125.00C    | 125.00C  |
| POWER  | 0.00     | 5.00     | 0.00       | 4.50     |

The D latch with an extra transmission gate added to control race conditions as shown in Figure 2.4 still results in a feedback voltage when T1 conducts due to clock lag. Thus, this circuit is also unusable for MACPITTS.

The master-slave flip-flop shown in Figure 2.4 can be operated as a single-phase or two-phase circuit [Ref. 4:pp. 213-215]. Two-phase operation will be considered in Section II B. For single-phase operation set  $\Phi 1 = \Phi 2$ . This circuit is immune to race conditions when configured as a single-phase or two-phase flip-flop, and is not as susceptible to feedback as the latch in Figure 2.1. This is a result of the first latch in the master-slave flip-flop in Figure 2.4 having transmission gate T3 as a load rather than an inverter as in Figure 2.1. Since a transmission gate has less capacitance, and thus less charge storage capability than an inverter, the clock lag which occurs during the clock transistion and

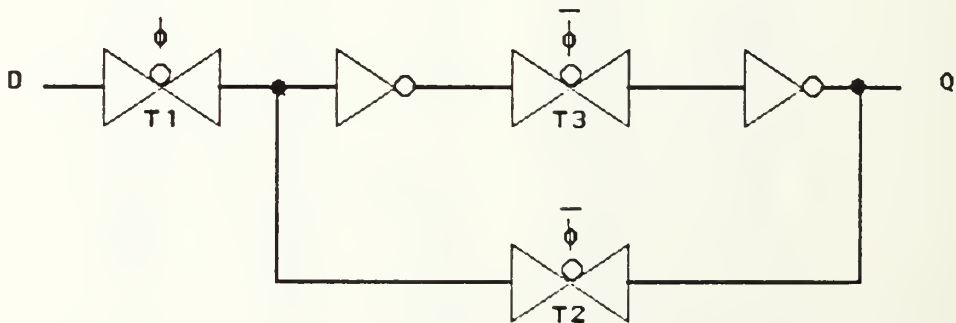


Figure 2.3 D Latch, Single Phase, Race Controllable

---

causes the feedback path results in less charge to drive the feedback path. This causes the feedback to have little effect on D since the drive is less.

SPICE simulations using nominal and worst case transistor parameters (Table 2.1) resulted in the following circuit times:

|            | NOMINAL         | WORST CASE     |
|------------|-----------------|----------------|
| CLOCK TO Q | $t_{pc}=3.1ns$  | $t_{pc}=3.9ns$ |
| DATA TO Q  | $t_{pd}=3.0ns$  | $t_{pd}=3.2ns$ |
| HOLD TIME  | $t_{sd}=-0.1ns$ | $t_{sd}=0.4ns$ |
| SETUP TIME | $t_{sc}=1.3ns$  | $t_{sc}=1.7ns$ |
| SKEW       | --              | $2S=0.1ns$     |

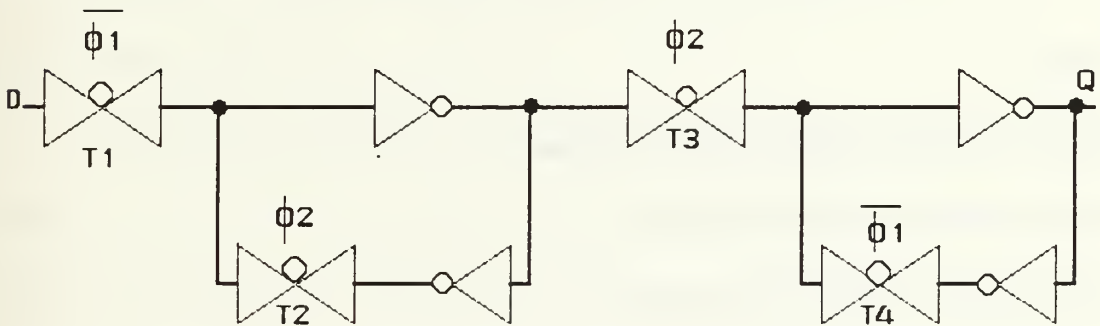


Figure 2.4 Master-Slave Flip Flop. For Single Phase  $\Phi 1 = \Phi 2$

CLOCK LAG : --  $G_i = 1.0ns$

PULSE WIDTH:  $w = 6.7ns$   $W = 7.0ns$

where,

$lpc$  = nominal delay time for clock to output

$lpd$  = nominal delay time for data to output

$lsd$  = nominal hold time

$lsc$  = nominal setup time

Upper case letters are worst case delay times.

Figures 2.5 and 2.6 show the simulation model and skew model used. Equations for the optimal clock period and pulse width are [Ref. 7:pp. 367] :

$$p = Lpc - \left( \frac{d - 2(Wt + 1)S - (Wt)Lsd + lpc + lsd}{Wt} \right) + D \quad (2.1)$$

$$w = Max \left( Lsc, 2S + \frac{(d - 2(Wt + 1)S + lpc + lsd)}{Wt} \right) \quad (2.2)$$

where,

$Wt$  = clock pulse width variation ( $W/w$ )

$W$  = maximum clock pulse width

$w$  = minimum clock pulse width

$D$  = maximum delay through combinational logic

$d$  = minimum delay through combinational logic.

The above values when inserted into the equations 2.1 and 2.2 yield:

$$p = 1.64ns - 0.95d + D$$

$$w = \text{Max}(1.7ns, 2.76ns + 0.95d)$$

$$= 2.76ns + 0.95d$$

An alternative D latch is shown in Figure 2.7 [Ref. 4:pp. 215-217]. This circuit resulted in race immune conditions when simulated using SPICE. As a static latch it operates well, but when configured as a flip-flop it requires 14 transistors more than the flip-flop in Figure 2.4.

SPICE simulations for the latch in Figure 2.7 resulted in nominal delay times for clock to Q of 3.9ns and data to Q of 4.1ns. Both of these times are greater than the delay times for the flip-flop in Figure 2.4. When configured as a flip-flop the delay times will even be greater. Although the single-phase clock with no complement is an ideal feature, the large circuit area required when configured as a flip-flop is not ideal. This along with the longer delay times makes this circuit undesirable for MACPITTS.

## B. TWO-PHASE CLOCKING

The master-slave flip-flop in Figure 2.4 is race immune [Ref. 4:pp. 213-215]. This circuit is even less susceptible to feedback than its single-phase counterpart due to the two-phases having more control over the feedback path. Detailed SPICE simulations for this circuit were not conducted as the delay times will only be relevant for the particular clock phase lag used in the simulations.

One disadvantage of this circuit is the number of clock lines that need to be routed. Since the circuit is two-phase, four clock lines will need to be routed, two for  $\Phi 1$  and  $\Phi 2$  and two for their complements. The extra area for routing is



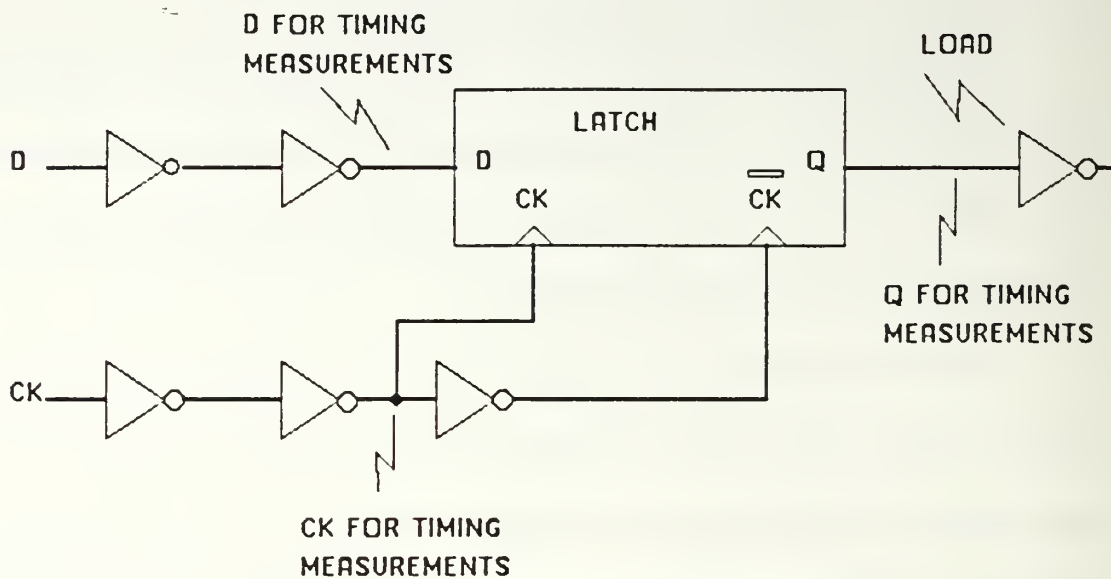


Figure 2.5 Simulation Model For All Circuits

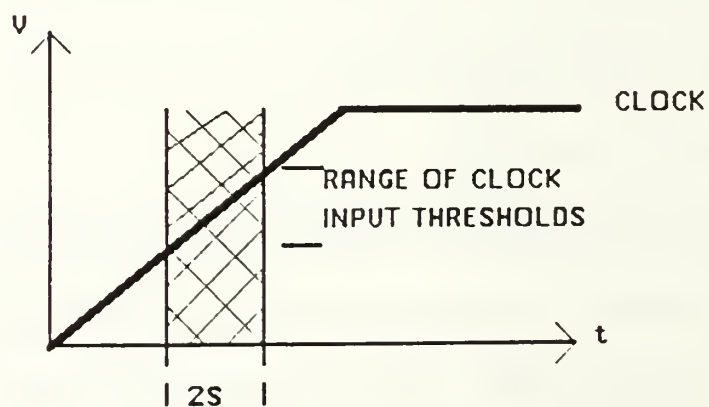


Figure 2.6 Circuit Skew Model

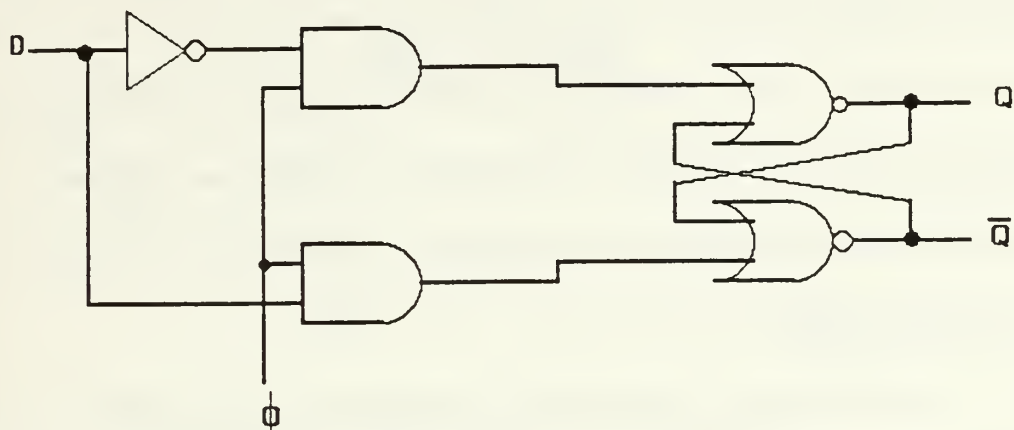


Figure 2.7a Static Single Phase D Latch Logic Diagram

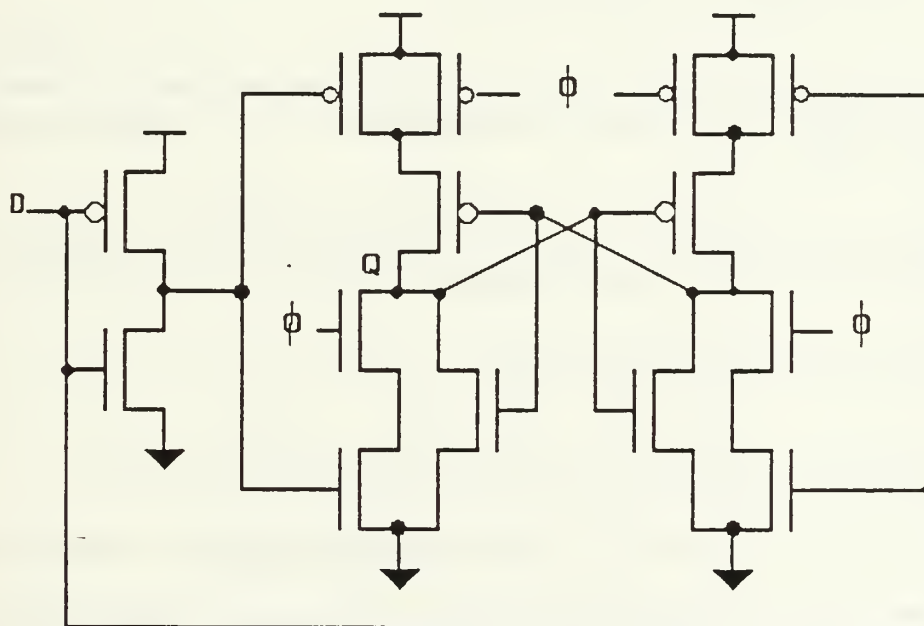


Figure 2.7b Static Single Phase D Latch Schematic

undesirable for MACPITTS but, when the single-phase case is considered, two circuits for the price of one can be obtained. Ideally, a switch could be inserted into MACPITTS software to select designs that operate as single- or two-phase. If the single-phase circuit operation is not reliable enough, MACPITTS could be re-executed with the switch set to re-configure the circuit as two-phase. The more reliable operation would be at the expense of added chip area due to the extra clock lines, however.

For comparison purposes with the single-phase version, SPICE simulations were generated using a non-overlapping clock and a fixed clock lag, T12 as shown in Figure 2.8. The value of T12 was calculated by:

- Conducting SPICE simulations to get the minimum clock pulse width for  $\Phi 1$  of 1.9ns required to latch the data.
- Using an estimate of  $2K\mu m$  for routing differences between  $\Phi 1$  and  $\Phi 2$ . Using first metal over field and a  $3\mu m$  wide metal path results in approximately 0.1ns delay.
- Using one inverter to generate the complement of  $\Phi 1$  resulting in a 0.6ns delay between  $\Phi 1$  and  $\Phi 1$  complement.
- Using a worst case skew of 0.1ns.
- Adding the delays in the above four items gives  $T12 = 2.7ns$ .

In an actual circuit T12 would probably be smaller, causing an overlap of  $\Phi 1$  and  $\Phi 2$ . This would prevent the inverter driving T2 in Figure 2.4 from fighting the gate that drives T1 when  $\Phi 1$  and  $\Phi 2$  are both low. SPICE simulations using nominal and worst case (Table 2.1) transistor parameters resulted in the following

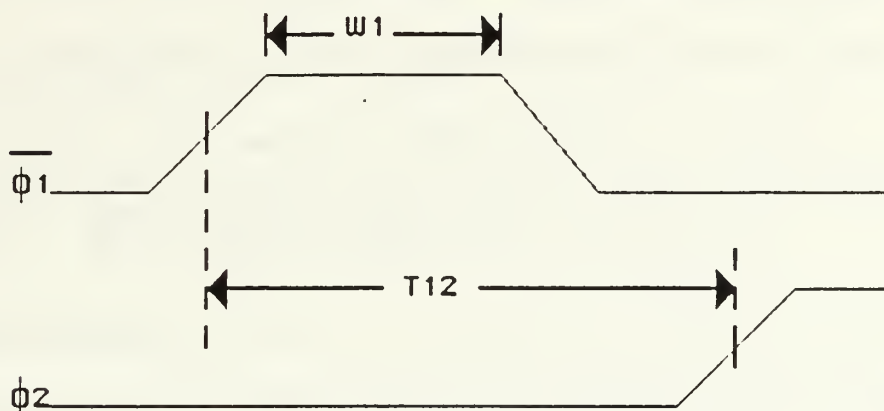


Figure 2.8 Two Phase Non-overlapping Clock

circuit times:

|                         | NOMINAL  | WORST CASE        |
|-------------------------|----------|-------------------|
| CLOCK1 TO T3: $l1_{pc}$ | $2.1ns$  | $L1_{pc} = 2.4ns$ |
| DATA TO T3 : $l1_{pd}$  | $1.5ns$  | $L1_{pd} = 1.6ns$ |
| CLOCK2 TO Q : $l2_{pc}$ | $3.5ns$  | $L2_{pc} = 4.4ns$ |
| T3 TO Q : $l2_{pd}$     | $1.5ns$  | $L2_{pd} = 1.6ns$ |
| HOLD TIME : $l1_{sd}$   | $-0.1ns$ | $L1_{sd} = 0.4ns$ |
| SETUP TIME : $l1_{sc}$  | $1.3ns$  | $L1_{sc} = 1.7ns$ |
| SKEW :                  | --       | $2S = 0.1ns$      |
| CLOCK LAG :             | --       | $Gi = 2.2ns$      |

$$PULSE\ WIDTH : w = 1.9ns \quad W = 2.2ns$$

See Figure 2.6 for skew model used. Equations for the two-phase optimal clock period and pulse width are [Ref. 7:p. 368] :

$$p = -d + (Wt - 1)(L1sc - L1sd - 2S) + L2pc + D \\ + (Wt)L1sd + 2(Wt + 1)S - l2pc - l1sd \quad (2.3)$$

$$w1 = \left( \frac{d + L2pd + L1pc - L2pc - 2S + l2pc + l1sd}{Wt} \right) \quad (2.4)$$

where the variables are the same as in the single-phase case and the subscripts 1 and 2 are used to distinguish between the phases. The above values when inserted into equations 2.3 and 2.4 yield:

$$p = 2.07ns - d + D$$

$$w1 = 2.33ns + 0.864d$$

based on T12 fixed at 2.17ns.

## C. APPLICATIONS

The equations for the minimum clock period  $p$  and minimum pulse width  $w$  ( $w1$ ) for the single-phase (two-phase) case can be used to calculate an approximate clock speed for a MACPITTS generated circuit. To do this, all that needs to be done is to:

- Generate the desired circuit layout using MACPITTS.
- Analyze the circuit using the CRYSTAL simulation package [Ref 5:pp. 297-319].
- Use the "critical" command to determine the critical path of the circuit.



- Add the worst case delay times for each organelle in the critical path. This generates D.
- Add 0.1ns delay to D for every 2K $\mu$ m of metal for signals in the critical path.
- Add the nominal delay times for each organelle in the critical path. This generates d.
- Insert the values of D and d found in the above items into the optimizing equations to find the maximum clock speed.

## D. CONCLUSIONS

The master-slave flip-flop in Figure 2.4 is ideally suited for MACPITTS. The possibility of configuring it as either a single-phase or two-phase structure opens the door for many different possibilities for MACPITTS. It allows a MACPITTS generated circuit to be operated internally as single-phase with an off chip single-phase clock, or the circuit can be configured with an internal two-phase clock and driven by an external single-phase clock, or even a two-phase internal clock and a two-phase external clock.

The race immune conditions of the flip-flop along with the short set up and delay times allows for a fast, reliable MACPITTS generated circuit. Thus, this is the circuit that will be used in MACPITTS.

### III. LAYOUT PHILOSOPHY

#### A. SCHEMATIC GENERATION

Since the organelles are designed in CMOS, schematic generation is an easy process. The p-channel and n-channel transistors can be represented as simple switches. See [Ref. 4:pp. 9-14] for a detailed explanation of switch representation. If two n-switches are placed in series, then the composite switch is on if both switches are on, that is, both n-channel transistor gate voltages are logical ones. This produces an AND function. The same is true for two p-channel transistors except they both conduct when the p-channel gate voltages are logical zeros.

If two n-switches are placed in parallel, then the composite switch is on if one or both switches are on, that is, one or both n-channel transistor gate voltages are logical ones. This produces an OR function. The same is true for two p-channel transistors except one or both p-channel gate voltages are logical zeros.

To implement compound functions in CMOS, all that needs to be done is to start with the n-channel pulldown structure and use a combination of series (AND) and parallel (OR) switch structures to represent the inverted expression. Once the n-side of the schematic is generated the complement of the switch structure is formed to represent the p-side. Wherever there exists a parallel combination of n-switches, this results in a series combination in the p-side. For a series combination of n-switches the p-side is implemented as a parallel

combination. The final step is to connect one side of the p-structure to Vdd, the other side to the output and one side of the n-structure to GND, the other side to the output.

## B. SPICE SIMULATIONS

Before the schematic can be used to layout an organelle the transistors in the circuit must be sized for proper drive and the circuit simulated to test for functionality and speed. This is done as a check to make sure that what is going to be generated on the CAD system is logically and electrically correct. Without this check a lot of time and money could be invested on a chip only to have non-functioning organelles.

SPICE was the only simulation tool used to evaluate all the organelles for functionality, transistor sizes, and to obtain propagation delays while ESIM [Ref. 5:pp. 19-22] was used as a second check to simulate the more complex organelles for functionality. MOSIS transistor parameters (Table 2.1) were used for the SPICE transistor models. The model used for all simulations is shown in Figure 2.5. All inputs are buffered to provide an ideal on-chip signal. Outputs are loaded with an inverter to provide a realistic load as would be seen by the organelle on a chip. The load inverter transistors are sized according to required fanout. The fanouts were selected for each organelle to be one and four. Loads with a fanout greater than four were not simulated as the rise and fall times are too great to be of any use for MACPITTS purposes. This is not to say that the organelles cannot

drive loads with a fanout greater than four; it means that fanouts greater than four are unsuited for MACPITTS.

### 1. Sizing Transistors

A minimum size scalable CMOS (SCMOS) transistor for a  $3\mu\text{m}$  minimum feature size process has a  $3.0\mu\text{m}$  width and a  $4.5\mu\text{m}$  length. Any circuit having an output with both n and p-channel transistors equal to these sizes is considered to have a drive of 1x. Since the p-channel mobility is one half that of the n-channel, all drives greater than 1x were designed to have their p-channel widths equal to twice their n-channel widths. That is, a 2x drive will have the p-channel width equal to  $9.0\mu\text{m}$  and the n-channel equal to  $4.5\mu\text{m}$ . For drives greater than 2x just multiply the 2x drive transistor widths by one half the desired drive to get the proper transistor widths. This will allow for nearly equal rise and fall times on all circuits with drives greater than one. For example, for a 6x drive transistor the width would be 3 times the 2x drive transistor.

Wherever possible circuits should be designed with minimum size transistors. This allows the organelle to be smaller and reduces loading on the organelle's driver. This is not always possible, however. Some circuits like NAND and NOR gates require larger transistors due to combinations of series and parallel transistors. To determine the correct transistor sizes  $R_{totalpchannel}$  should equal  $\frac{1}{2} \times R_{totalnchannel}$  where,  $R_{total} = R_1 + R_2 + \dots + R_N$  for series transistors and

$$R_{total} = \left( \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \dots + \frac{1}{R_1}} \right) \text{ for parallel transistors. By decreasing } R_{total} \text{ the}$$

output drive can be increased. Therefore, increasing transistor widths will decrease  $R_{total}$ . After determining  $R_{total}$  several simulations should be generated to fine tune the transistor sizes to obtain equal rise and fall times. This is not always possible, however, since transistor widths are on a grid in the CAD system and thus must be multiples of this grid. Also, it is not always desirable to have equal rise and fall times if the increase in transistor area required to achieve this is excessive. These are considerations that must be evaluated when simulating the circuits.

## 2. Circuit Functionality

Circuit functionality was tested using SPICE for all organelles and ESIM for a select few as a second check. A timing diagram was generated by hand to determine the correct circuit function. This timing diagram included all the entries in the truth tables to ensure a complete functionality check of the organelle. Once this is completed, the SPICE pulse function can be used to represent the input timing waveforms in the SPICE input file. After the simulation is completed its output waveforms should logically match the hand generated ones. If the two agree then the circuit is logically correct.

## 3. Propagation Delays

Propagation delays are determined from the SPICE functionality simulations. The propagation delay for a falling output ( $t_{df}$ ) and a rising output



( $t_{dr}$ ) are obtained by taking the time difference between the 50% point of the input waveform and the 50% point of the output waveform. The rise and fall times ( $t_r$  and  $t_f$  respectfully) are obtained by taking 10% to 90% of full swing of the output waveforms.

### C. STICK DIAGRAMS

Stick diagrams were used initially for the organelle layouts. The idea is to have a simple representation of the organelle on paper before using MAGIC to capture the layout. The stick diagrams allow the designer to make several quick layouts on paper in order to select the most efficient and smallest layout. It is best to use the same color scheme as MAGIC (red for poly, blue for first metal, etc.) to avoid confusion later on. The stick diagrams need not be totally correct in following MAGIC design rules. The idea is to provide quick, simple representations of the organelle as seen on the MAGIC terminal. If there are design-rule errors in the stick diagrams MAGIC informs the user during layout and they can be corrected at that time.

### D. MAGIC USAGE FOR STANDARD CELLS

As mentioned in chapter one, MAGIC was used extensively in cell layout. The MAGIC output style used for the layouts was  $\lambda = 1.5$  (gen). This is a generic process in which scalable rules apply to P-well as well as N-well and twin tub processes. This will generate Caltech Intermediate Format files for the MOSIS SCMOS technology with a  $3.0\mu\text{m}$  minimum feature size [Ref. 5:p. 295].



The design rules in [Ref. 5:pp. 285-296] were used to implement the layout of the organelles. Minimum transistor sizes have a  $3.0\mu\text{m}$  width and a  $4.5\mu\text{m}$  length. In addition to the MAGIC design rules, the following design rules were also used to layout the organelles:

- All I/O points are on first metal with inputs on one edge of the organelle and outputs on the opposite edge.
- First metal and poly are used for signal and power routing within organelles.
- External CLOCK, Vdd, and GND connections are on second metal only, and run the full length of the organelle perpendicular to I/O. No other second metal is used in the organelles.
- All external connections to I/O, CLOCK, Vdd, and GND end at least 5 units past all transistors.
- All external connections to I/O, CLOCK, Vdd, and GND end at least 4 units past all substrate contacts.
- All external connections to I/O, CLOCK, Vdd, and GND end at least 2 units past all poly.
- All external connections to I/O, CLOCK, Vdd, and GND end at least 2 units past first metal that is not an I/O point.
- All external connections to I/O, CLOCK, Vdd, and GND end at least 2 units past second metal that is not a CLOCK, Vdd, or GND point.

The above design rules were set in order to allow identical organelles to abut. The I/O, CLOCK, Vdd, and GND points determine the boundaries for the organelles. Thus, identical cell boundaries can touch without causing design rule violations. This is useful, for example, for adder organelle applications. For an  $n$ -bit adder,  $n$  adder organelles are simply stacked. All Vdd and GND busses line up

and run the entire length of the n-bit adder, and no design rule violations should occur.

For cells that are not identical, care must be taken when placing the organelles. The boundaries can still touch but, because CLOCK, Vdd, and GND points may no longer line up, second metal design rules must be followed to ensure there are not any violations. The same is true for I/O points and first metal design rules.

## E. CHECKING LAYOUTS

Checking layouts is accomplished in two parts. The first part is done while the layout is being generated. It involves following MAGIC's design rules to layout the organelles. If followed correctly the white dots indicating design rule violations will not appear on the screen. If the white dots do not appear on the screen then the first part of the check is completed.

The second part of the check involves verification of the organelle. While in MAGIC with the organelle displayed on the screen type:

```
:extract
```

Then under the UNIX operating system type:

```
>ext2sim fn
```

```
>sim2spice fn
```

The second command generates fn.sim file that is used for ESIM simulations. The third command generates a SPICE input file of the layout. The SPICE input file

can be used to generate a schematic by hand. If this schematic matches the schematic used to generate the layout then the layout is topologically correct.

## IV. APPLICATIONS

### A. CIRCUIT SIZE COMPARISON

A comparison of the areas of selected SCMOS organelles and the MACPITTS NMOS organelles was conducted. Since static NMOS requires  $N + 1$  transistors and static CMOS requires  $2N$  transistors, it is reasonable to assume that CMOS requires approximately twice the chip area as NMOS (assuming the same layout style is used for both technologies). However, when layout styles among the technologies differ, area comparisons are not as simple because of the many variables introduced into the comparison. For example, SCMOS organelles using a hierarchical layout style will be more than twice the area of an NMOS organelle utilizing a standard layout, since CMOS is approximately twice the area of NMOS and hierarchical layouts result in larger layouts than standard layouts. The hierarchical layout style results in a larger layout than the standard layout style due to the fixed dimensions of the building blocks used in a hierarchical layout. The fixed dimensions cause all of the routing connecting the building blocks together to lay outside these fixed boundaries, thus increasing the overall area.

Since MACPITTS NMOS organelles use a hierarchical layout style and the SCMOS organelles use a standard layout style the area differences had to be calculated since no rule of thumb exists for layout style area comparisons. The

area measurements for a few typical organelles resulted in the following:

| ORGANELLE    | SCMOS AREA ( $\lambda^2$ ) | NMOS AREA ( $\lambda^2$ ) | % CHANGE |
|--------------|----------------------------|---------------------------|----------|
| 2 INPUT XNOR | 4536.8                     | 3538                      | 22       |
| 2 INPUT NAND | 1946.8                     | 661.5                     | 66       |
| 2 INPUT NOR  | 2005.3                     | 742                       | 63       |
| 2 INPUT OR   | 2725.9                     | 1375                      | 49       |
| 2 INPUT AND  | 2488                       | 1225                      | 50       |

Since SCMOS areas are expected to be approximately double those for the comparable NMOS circuits, the above measurements show that the NMOS organelles are very inefficient layouts due to the inherent limitations of the hierarchical layout style.

## B. SIMULATION RESULTS

Although SPICE was the only simulation tool used to simulate all the organelles in the SCMOS library, another simulation package was also investigated for its usefulness to VLSI design simulation. ESIM, an event-driven switch-level simulator [Ref. 5:pp. 19-22], was used to simulate a selected group of organelles. The organelles selected were chosen on the basis of clocking strategy used (single-phase or two-phase) and transmission gates used (whether present or not). The simulations included a 2 to 1 MUX, a 2 input NAND gate, a single phase D flip-flop, and a two phase D flip-flop.

Circuits that involved transmission gates with no clocking mechanisms, such as the 2 to 1 MUX, and circuits that did not contain transmission gates,

such as the 2 input NAND gate, work reasonably well when simulated with ESIM. All possible input combinations for the 2 to 1 MUX and the 2 input NAND gate were used in the simulation which generated the correct output for each input. In addition, several different circuits were constructed involving the MUX feeding one input of the NAND gate or the NAND gate feeding one input of the MUX. The simulations of these circuits also produced the correct outputs.

Circuits that use non-overlapping clocking mechanisms, such as the two-phase D flip-flop, simulate correctly using ESIM. A problem arises when an overlapping clocking mechanism, such as in the single phase D flip-flop, is used. Because of the overlap ESIM will generate unknowns for all nodes that are clocked and all nodes that follow a clocked node. The problem with overlapping clocks was verified by overlapping the clocks in the two phase D flip-flop which generated the same unknowns as the single phase D flip-flop.

### C. APPLICATION EXAMPLES

Several applications for the organelle library will be discussed next. Besides being used as the SCMOS organelle library for MACPITTS the organelles can also be used to generate hand crafted layouts.

The one bit adder organelle can be used to generate an n-bit adder. This is easily accomplished by abutting n-adder organelles so that the power rails line up. Once this is done the  $C_{OUT}$  of bit n is simply connected via first metal to  $C_{IN}$  of bit  $n + 1$ .



The look ahead carry organelle is a four stage static look ahead carry. However, only stage three of the look ahead carry was constructed. This is due to the fact that stages one, two, and four can be obtained with relatively few organelles. Stage one is obtained by:

$$COUT1 = G1 + P1 \cdot CIN$$

where,

CIN = carry in

P1 to PN = propagatel to propagateN

G1 to GN = generatel to generateN

This requires only a two input OR gate and a two input AND gate. Stage two can be obtained by setting  $G3 = 0$  and  $P3 = 1$  in stage three since:

$$COUT2 = G2 + P2(G1 + P1 \cdot CIN)$$

$$COUT3 = G3 + P3(G2 + P2(G1 + P1 \cdot CIN))$$

Stage four can be obtained by:

$$COUT4 = G4 + P4 \cdot COUT3$$

This requires only a two input OR gate, a two input AND gate, and the look ahead carry organelle. The look ahead carry organelle was constructed using compound gates. That is, the organelle was implemented as one function rather than as a cascade of logic gates. By using compound gates the organelle speed was increased to the point where it is expected to be as fast as a two level cascade look ahead carry.

Other applications include using the two input XNOR gate as an equality organelle since  $\overline{A \oplus B} = 1$  only when  $A = B$ . The two input XOR gate can be used as an inequality organelle since  $A \oplus B = 1$  only when  $A \neq B$ . These are just some of the many applications that can be generated using the organelle library.

## V. CONCLUSIONS

The goal of this thesis was to develop a standard CMOS library for use in converting the MACPITTS silicon compiler from NMOS to CMOS technology. The cells were designed using a bit slice approach (organelle) for easy integration into the MACPITTS software architecture. The main result of the thesis is the development of enough organelles to allow for a CMOS conversion of MACPITTS and to allow for hand crafted VLSI layouts using the organelles.

It was shown that the three phase clocking scheme used in the NMOS MACPITTS was too conservative. Several clocking schemes were investigated. A two phase clocking scheme was selected as being just as reliable as the three phase clocking scheme only requiring fewer transistors for the circuits. This was the approach used in developing all clocked cells. Additionally, a single phase flip flop was developed for MACPITTS for incorporation into those designs where clock skew is not a strict requirement.

The simulations conducted resulted in delay times being tabulated for each cell along with demonstrating that the cells are functionally correct. The tabulated delay times allow a designer to calculate propagation delay and clock speed for a particular circuit.

The more than twenty cells constructed for the library are just a start for a standard CMOS library. Many more possible cells can be added to allow the

library to be a highly useful tool to the VLSI designer and to increase the capabilities of MACPITTS.

Some recommended additions to the library include shift register organelles, a stackable one bit multiplier organelle and, a four to one and eight to one multiplexer. Test functions for these organelles would have to be generated and included in MACPITTS for the organelles to be used by the compiler. These additional organelles along with the existing organelles would enable a designer to generate any number of VLSI circuits, which normally take many man-months to design and layout, in just a few hours.

## APPENDIX . ORGANELLES

| GLOSSARY          |   |
|-------------------|---|
| ORGANELLE         | FUNCTION  |
| ADDER             | ONE BIT ADDER                                       |
| AND2              | 2 INPUT AND GATE                                    |
| BUFFER1X          | NON-INVERTING BUFFER,<br>MINIMUM DRIVE              |
| BUFFER1X-4X       | NON-INVERTING BUFFER,<br>4X DRIVE                   |
| DFF1PHASE         | MASTER-SLAVE D FLIP FLOP,<br>SINGLE PHASE, NO CLEAR |
| DFF2PHASE         | MASTER-SLAVE D FLIP FLOP,<br>TWO PHASE, NO CLEAR    |
| INV1X             | INVERTER. MINIMUM DRIVE                             |
| INV4X             | INVERTER. 4X DRIVE                                  |
| INV8X             | INVERTER. 8X DRIVE                                  |
| LOOK-AHEAD-CARRY4 | STAGE 3 OF A 4 STAGE STATIC<br>LOOK AHEAD CARRY     |
| MUX2-1            | 2 TO 1 MULTIPLEXER                                  |
| NAND2             | 2 INPUT NAND GATE                                   |
| NAND3             | 3 INPUT NAND GATE                                   |
| NAND4             | 4 INPUT NAND GATE                                   |
| NOR2              | 2 INPUT NOR GATE                                    |
| NOR3              | 3 INPUT NOR GATE                                    |
| NOR4              | 4 INPUT NOR GATE                                    |
| OR2               | 2 INPUT OR GATE                                     |
| ORANDINV3         | 3 INPUT OR AND INVERT GATE                          |
| XNOR2             | 2 INPUT XNOR GATE                                   |
| XOR2              | 2 INPUT XOR GATE                                    |

- $t_{dr}$  is the propagation delay for a rising output which is obtained by taking the time difference between the 50% point of the input waveform and the 50% point of the output waveform.
- $t_{df}$  is the propagation delay for a falling output which is obtained by taking the time difference between the 50% point of the input waveform and the 50% point of the output waveform.
- $t_r$  is the rise time for the output waveform which is obtained by taking 10% to 90% of full swing of the output waveform.
- $t_f$  is the fall time for the output waveform which is obtained by taking 10% to 90% of full swing of the output waveform.



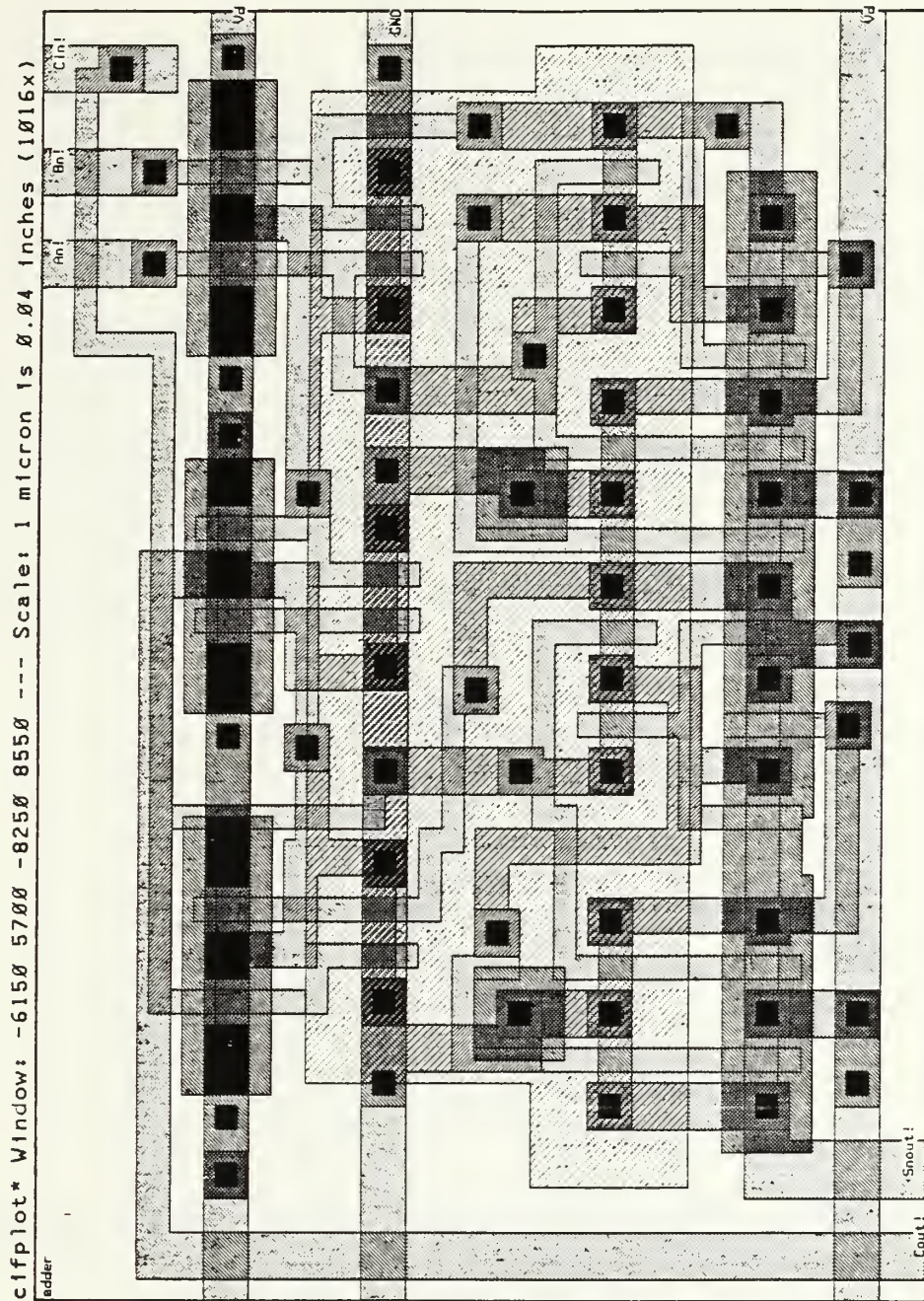


Figure A.1a Adder Cif Plot

---

| PARAMETER             | FANOUT | $t_{dr}$ | $t_{df}$ | $t_r$ | $t_f$ |
|-----------------------|--------|----------|----------|-------|-------|
| An to $S_{out}$       | 1      | 3.3ns    | 3.8ns    | 1.6ns | 1.4ns |
|                       | 4      | 4.1ns    | 4.2ns    | 3.5ns | 2.1ns |
| Bn to $S_{out}$       | 1      | 3.1ns    | 3.9ns    | 1.6ns | 1.4ns |
|                       | 4      | 4.0ns    | 4.3ns    | 3.5ns | 2.1ns |
| $C_{in}$ to $S_{out}$ | 1      | 1.7ns    | 2.7ns    | 1.6ns | 1.4ns |
|                       | 4      | 2.6ns    | 3.7ns    | 3.5ns | 2.1ns |
| An to $C_{out}$       | 1      | 2.0ns    | 1.9ns    | 1.5ns | 1.4ns |
|                       | 4      | 2.7ns    | 2.5ns    | 3.7ns | 2.9ns |
| Bn to $C_{out}$       | 1      | 1.7ns    | 1.6ns    | 1.5ns | 1.3ns |
|                       | 4      | 2.3ns    | 2.2ns    | 3.7ns | 2.9ns |
| $C_{in}$ to $C_{out}$ | 1      | 1.5ns    | 1.5ns    | 1.8ns | 1.4ns |
|                       | 4      | 2.1ns    | 2.2ns    | 3.5ns | 2.9ns |

| TRUTH TABLE |    |          |           |           |
|-------------|----|----------|-----------|-----------|
| An          | Bn | $C_{in}$ | $C_{out}$ | $S_{out}$ |
| 0           | 0  | 0        | 0         | 0         |
| 1           | 0  | 0        | 0         | 1         |
| 0           | 1  | 0        | 0         | 1         |
| 1           | 1  | 0        | 1         | 0         |
| 0           | 0  | 1        | 0         | 1         |
| 1           | 0  | 1        | 1         | 0         |
| 0           | 1  | 1        | 1         | 0         |
| 1           | 1  | 1        | 1         | 1         |

- Stackable one bit adder. N bit adder formed by stacking N organelles.

Figure A.1b Adder Timing Data

---

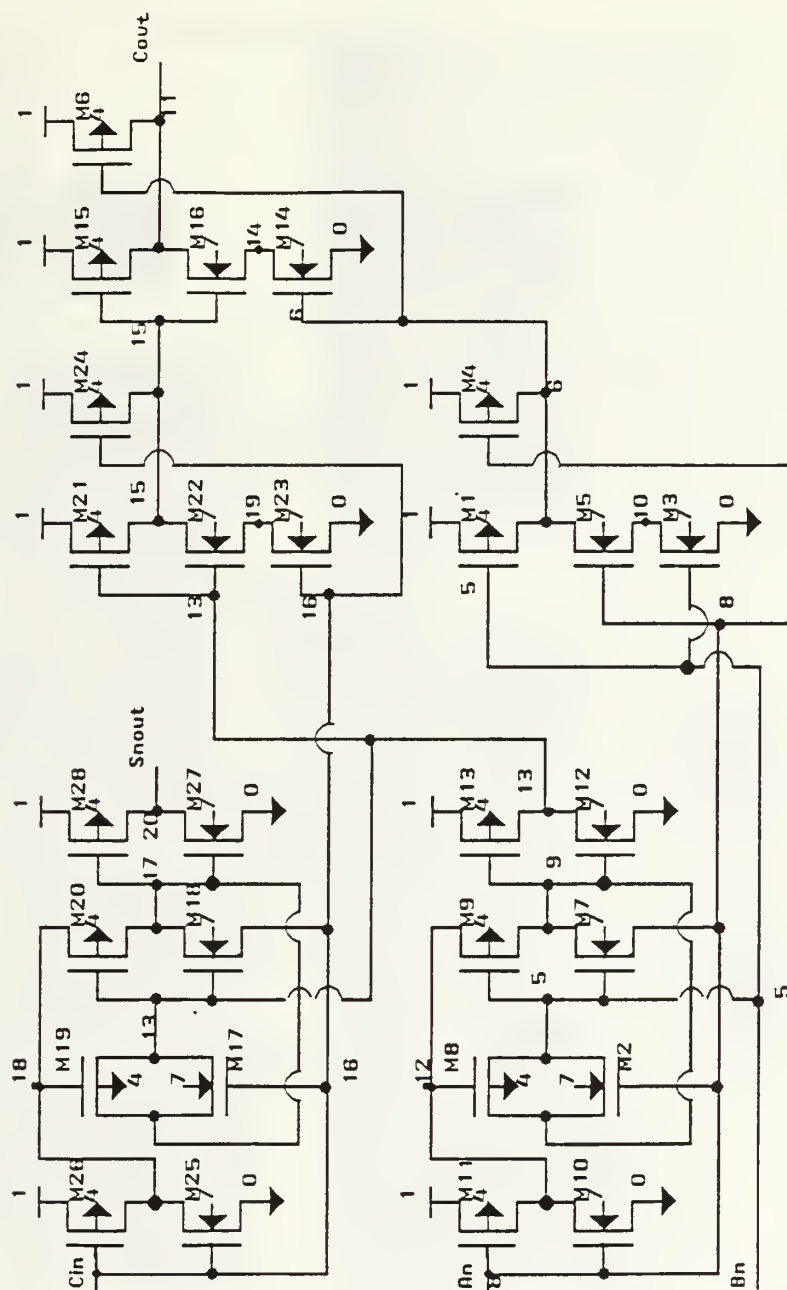


Figure A.1c Adder Schematic



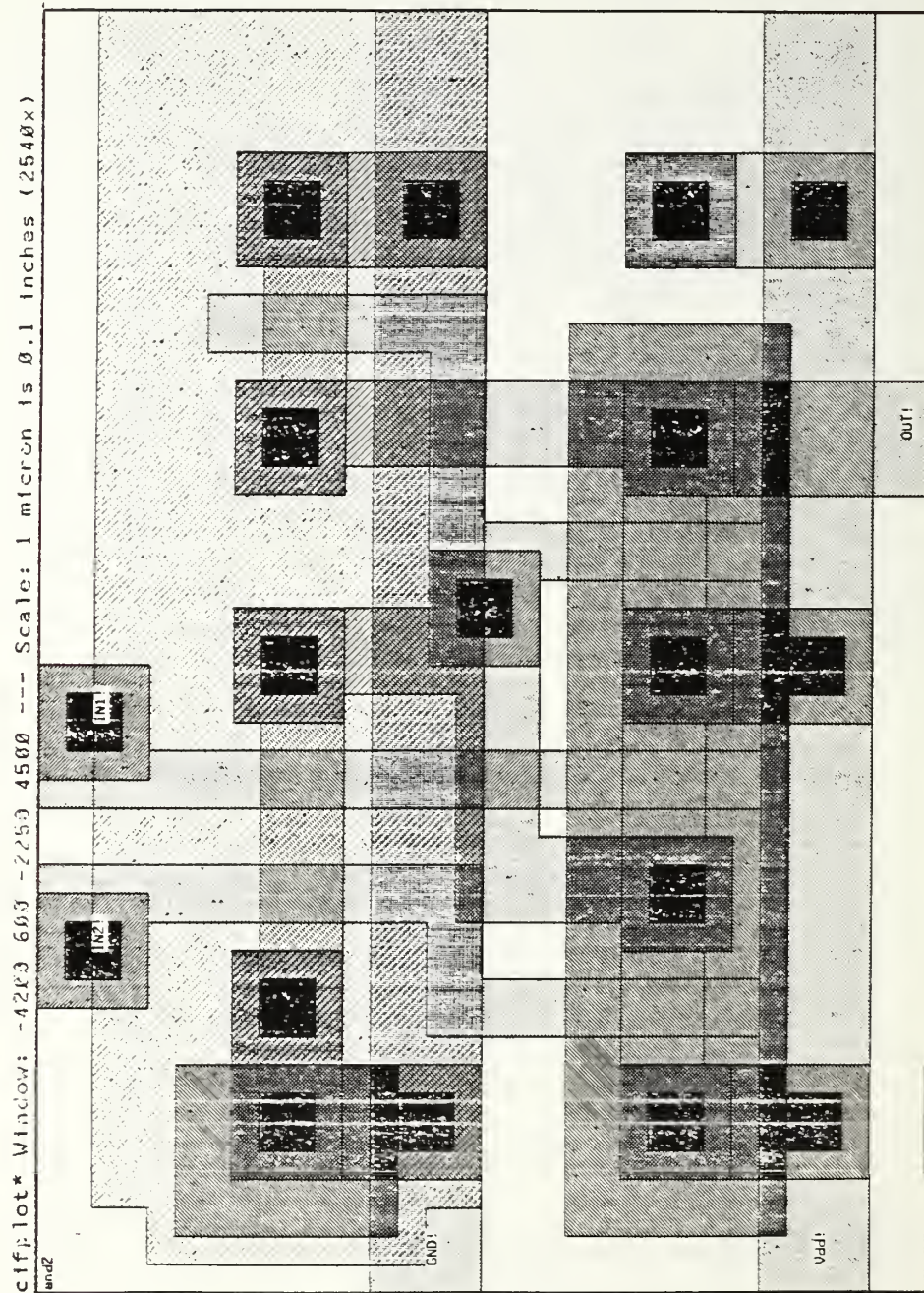


Figure A.2a And2 Cif Plot

---

| INPUT | FANOUT | $t_{dr}$ | $t_{df}$ | $t_r$ | $t_f$ |
|-------|--------|----------|----------|-------|-------|
| IN1   | 1      | 1.5ns    | 1.4ns    | 1.3ns | 0.9ns |
|       | 4      | 3.2ns    | 2.4ns    | 3.4ns | 1.7ns |
| IN2   | 1      | 1.7ns    | 1.5ns    | 1.3ns | 0.9ns |
|       | 4      | 3.3ns    | 2.4ns    | 3.4ns | 1.7ns |

| TRUTH TABLE |     |     |
|-------------|-----|-----|
| IN1         | IN2 | OUT |
| 0           | 0   | 0   |
| 1           | 0   | 0   |
| 0           | 1   | 0   |
| 1           | 1   | 1   |

Figure A.2b And2 Timing Data

---

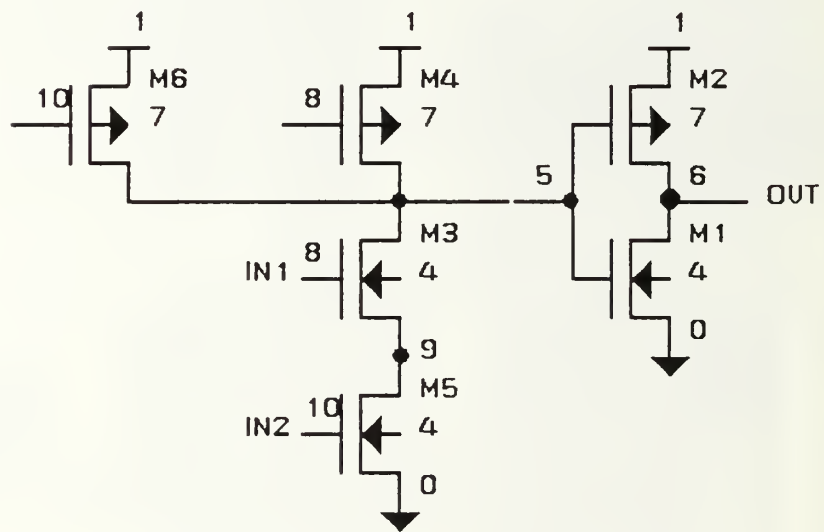


Figure A.2c And2 Schematic



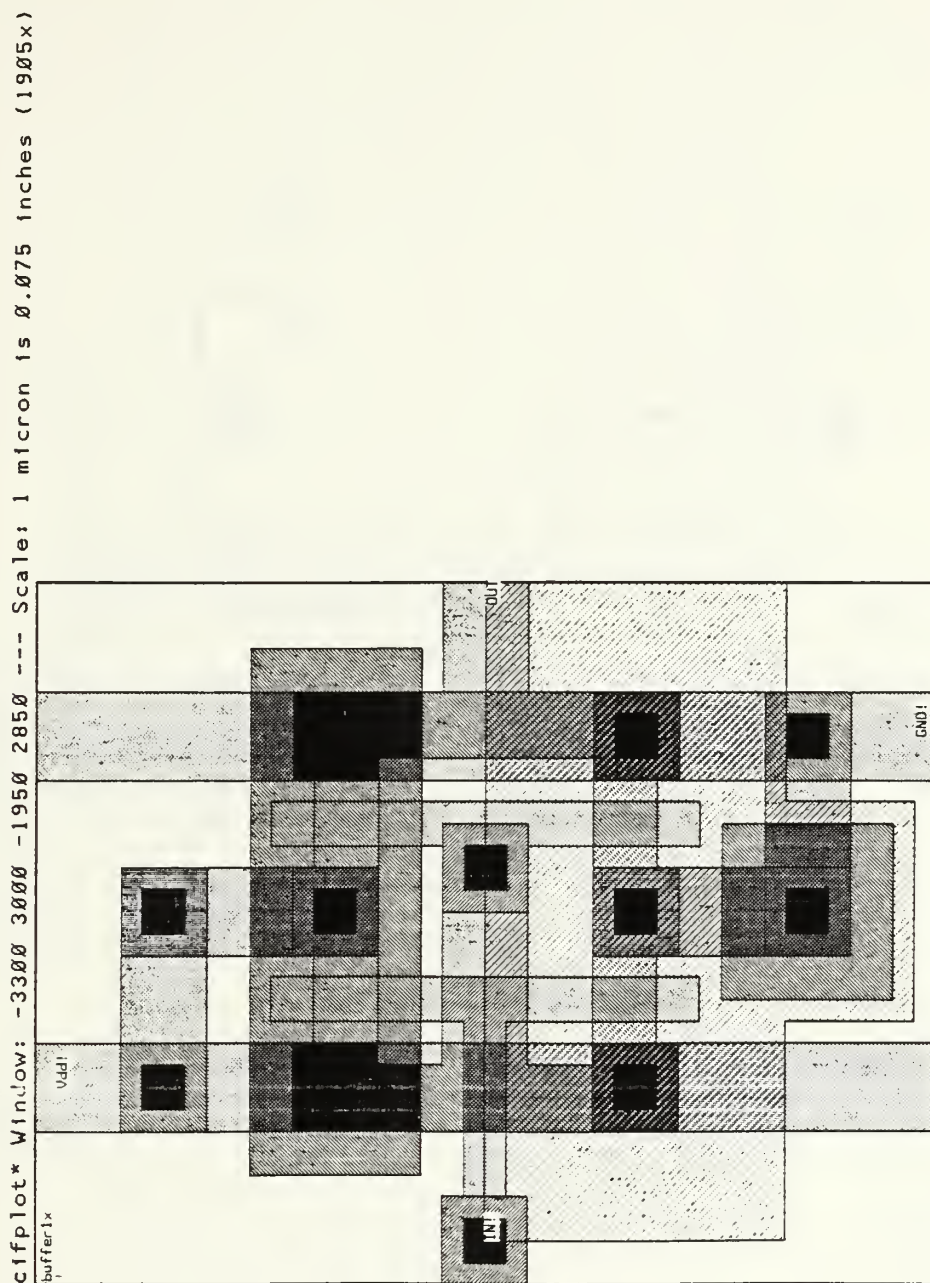


Figure A.3a Buffer1x Cif Plot

---

| FAN OUT | $t_{dr}$ | $t_{df}$ | $t_r$ | $t_f$ |
|---------|----------|----------|-------|-------|
| 1       | 1.3ns    | 1.3ns    | 1.3ns | 0.8ns |
| 4       | 2.2ns    | 1.75ns   | 3.2ns | 1.6ns |

| TRUTH TABLE |     |
|-------------|-----|
| IN          | OUT |
| 0           | 0   |
| 1           | 1   |

Figure A.3b Buffer1x Timing Data

---

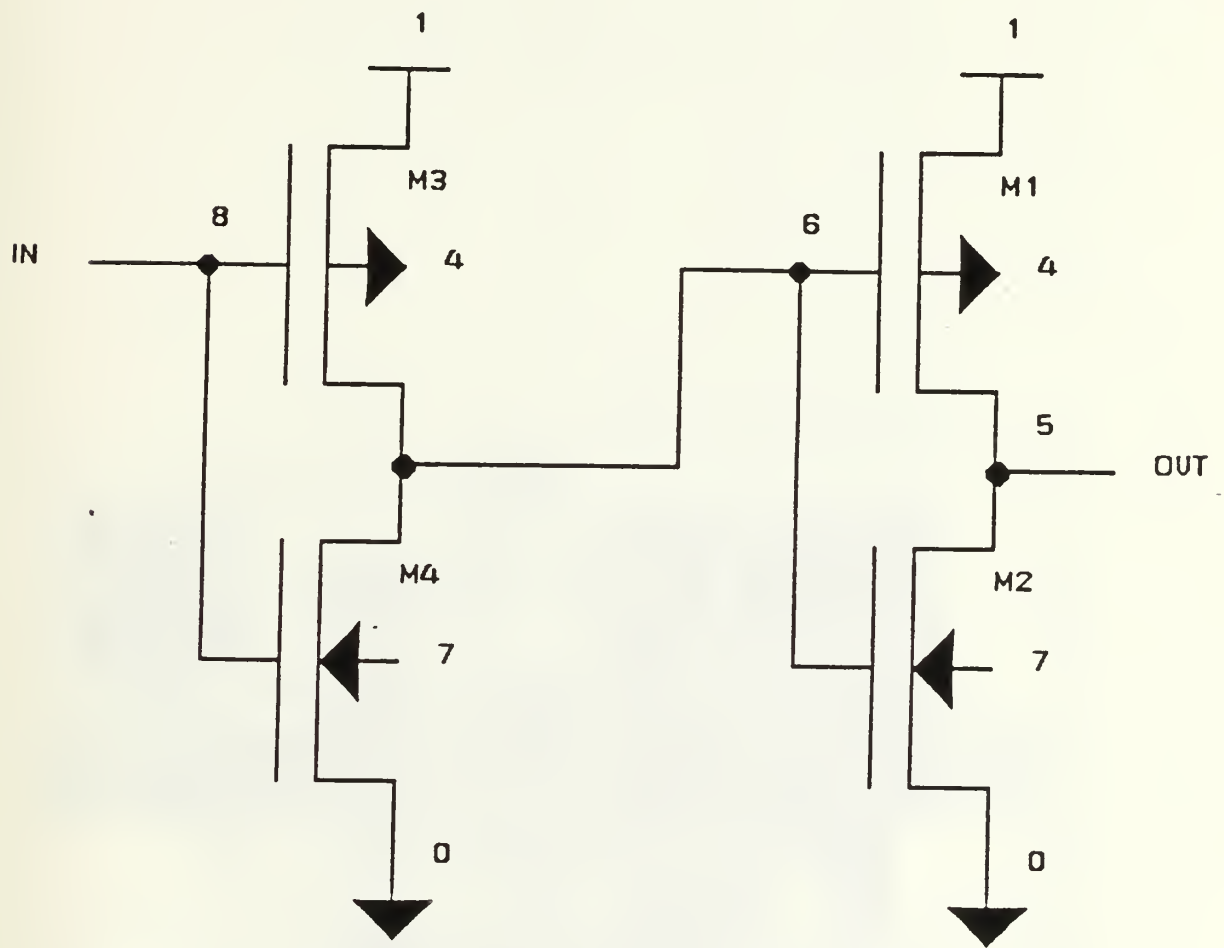


Figure A.3c Buffer1x Schematic

cifplot\* Window: -3750 3300 -1950 2850 --- Scale: 1 micron is 0.075 inches (1905x)

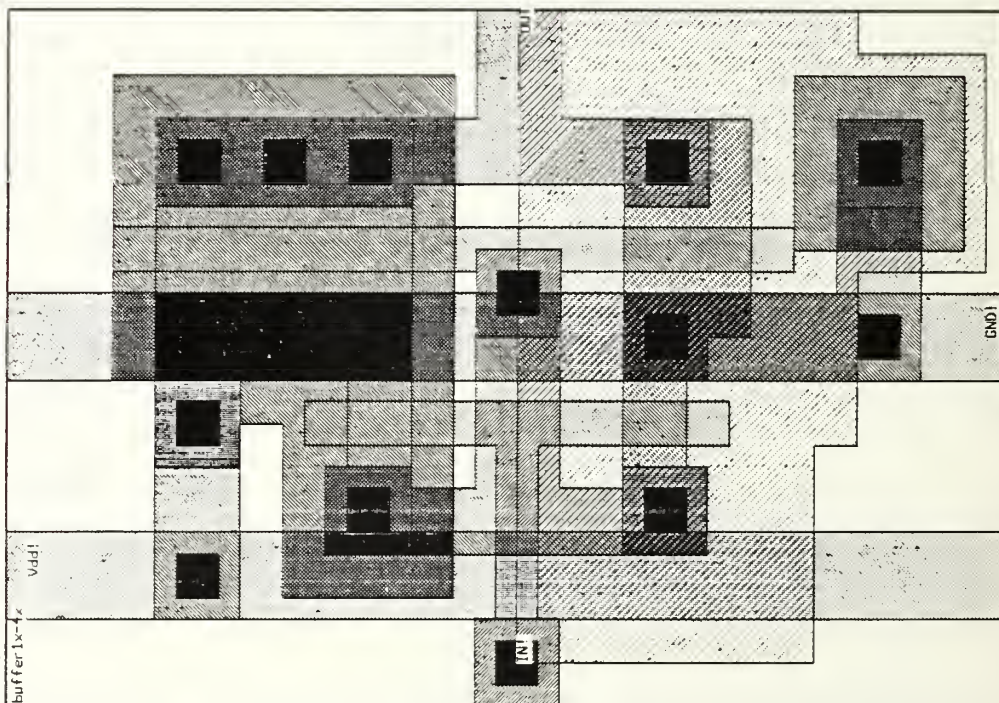


Figure A.4a Buffer1x-4x Cif Plot

---

| FAN OUT | $t_{dr}$ | $t_{df}$ | $t_r$ | $t_f$ |
|---------|----------|----------|-------|-------|
| 0.25    | 1.4ns    | 2.2ns    | 1.5ns | 1.2ns |
| 1       | 1.7ns    | 2.5ns    | 1.5ns | 1.3ns |
| 4       | 2.6ns    | 3.4ns    | 3.5ns | 3.2ns |

| TRUTH TABLE |     |
|-------------|-----|
| IN          | OUT |
| 0           | 0   |
| 1           | 1   |

Figure A.4b Buffer1x-4x Timing Data

---

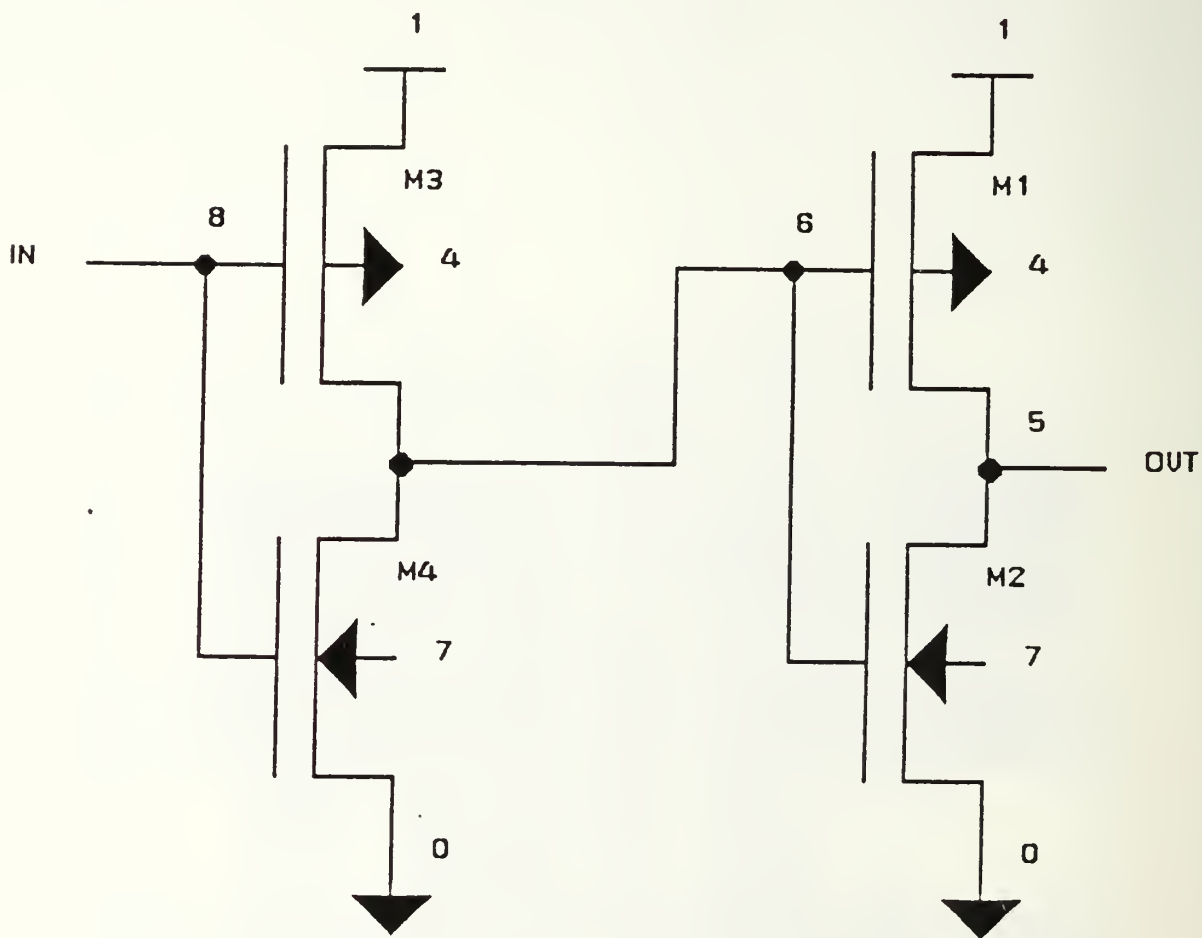


Figure A.4c Buffer1x-4x Schematic



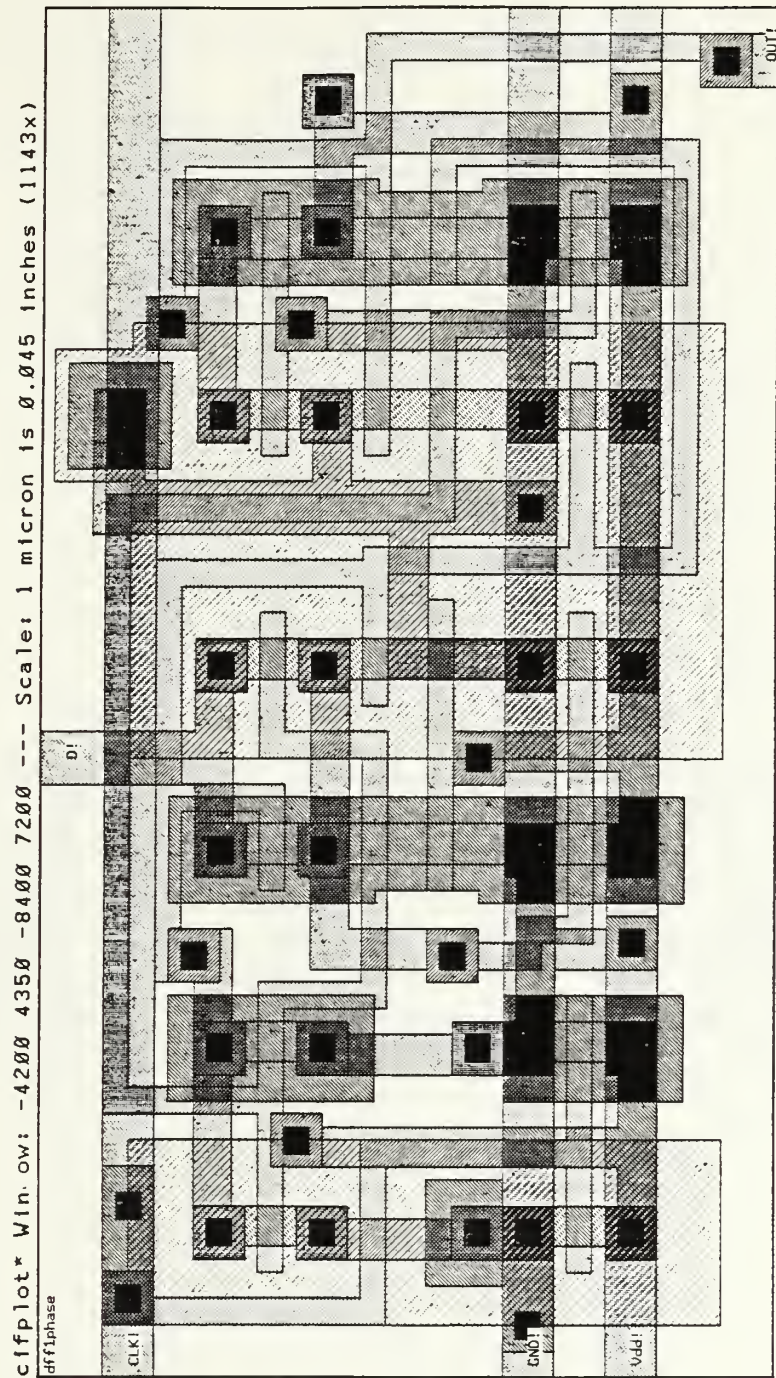


Figure A.5a Dff1phase Cif Plot

---

| PARAMETER  | $t_d$  |
|------------|--------|
| CLOCK TO Q | 3.1ns  |
| DATA TO Q  | 3.0ns  |
| HOLD TIME  | -0.1ns |
| SETUP TIME | 1.3ns  |

| TRUTH TABLE |      |       |
|-------------|------|-------|
| CLOCK       | DATA | Q     |
| RISING      | 0    | 0     |
| RISING      | 1    | 1     |
| LOW         | X    | $Q^+$ |

- Simulation conducted for a fanout of one only. This is equivalent to a fanout of two for an inv1x organelle due to one fanout for the unit load and one fanout for the feedback inverter in the last latch (see Figure A.5c). To obtain times for greater fanouts simply interpolate the time for a fanout of two for the inv1x organelle, subtract this from the desired parameter of dff1phase organelle to obtain the base delay, interpolate inv1x for the desired fanout plus one and add this to the base delay to get the desired delay times. Hold times and setup times are independent of fanout.

Figure A.5b Dff1phase Timing Data

---

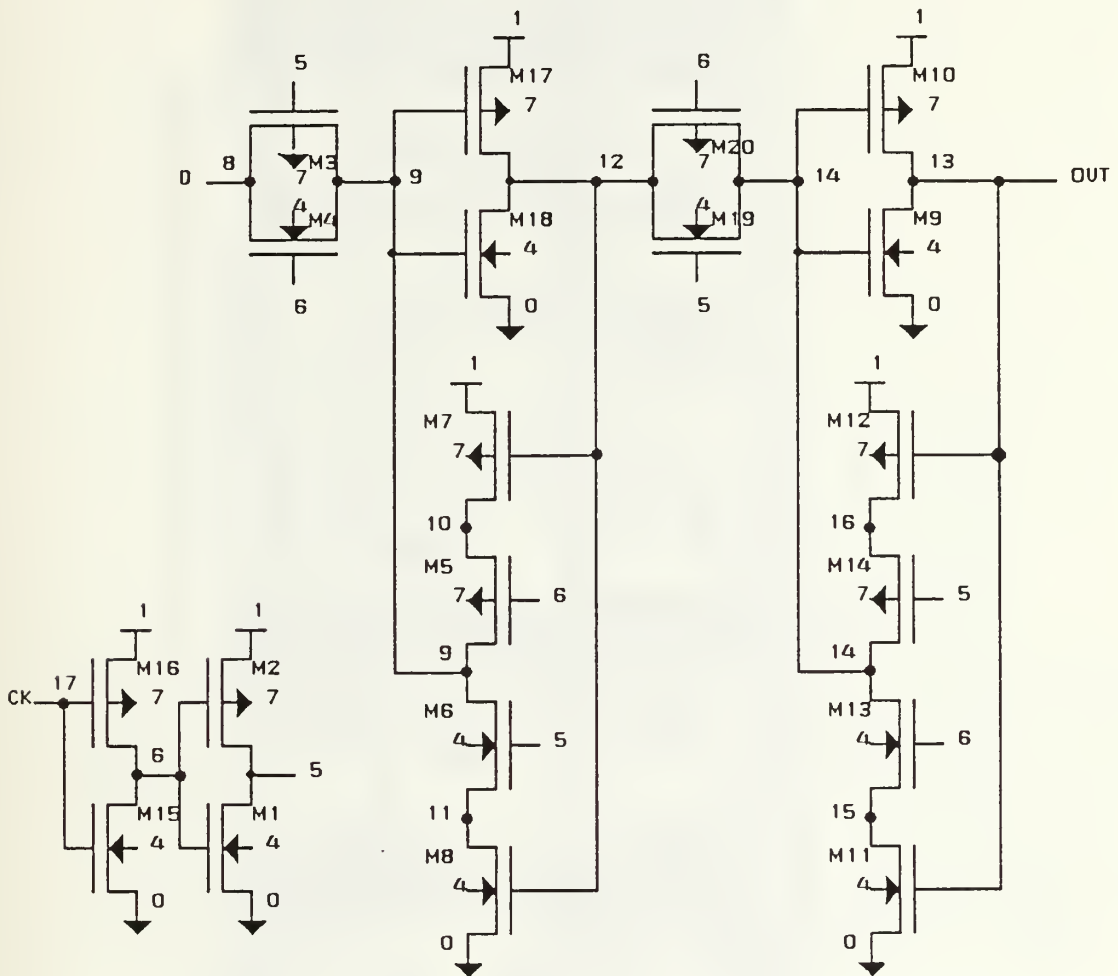


Figure A.5c Dff1phase Schematic



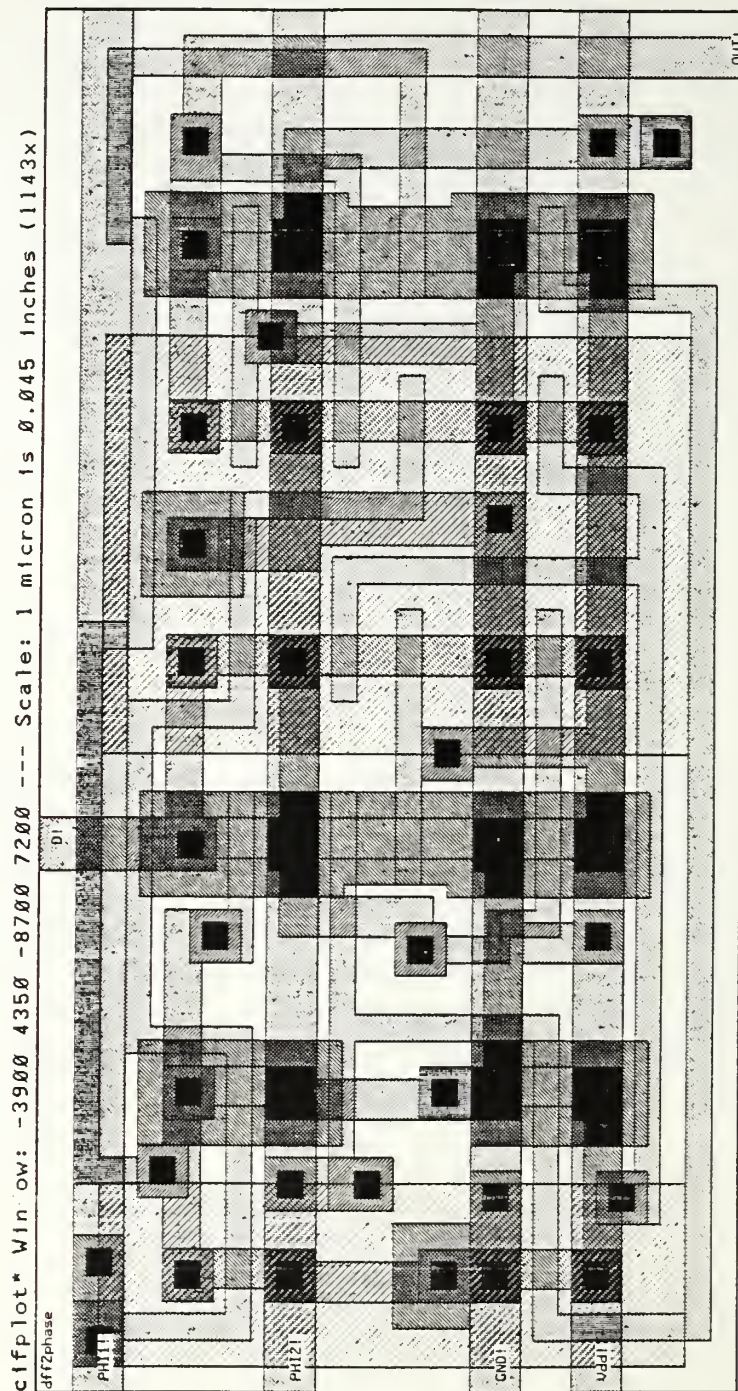


Figure A.6a Dff2phase Cif Plot

---

| PARAMETER    | $t_d$  |
|--------------|--------|
| CLOCK1 TO T3 | 2.4ns  |
| CLOCK2 TO Q  | 3.5ns  |
| DATA TO T3   | 1.5ns  |
| T3 TO Q      | 1.5ns  |
| HOLD TIME    | -0.1ns |
| SETUP TIME   | 1.3ns  |

| TRUTH TABLE |      |       |
|-------------|------|-------|
| CLOCK       | DATA | Q     |
| RISING      | 0    | 0     |
| RISING      | 1    | 1     |
| LOW         | X    | $Q^+$ |

- Simulation conducted for a fanout of one only. This is equivalent to a fanout of two for an invl<sub>x</sub> organelle due to one fanout for the unit load and one fanout for the feedback inverter in the last latch (see Figure A.6c). To obtain times for greater fanouts simply interpolate the time for a fanout of two for the invl<sub>x</sub> organelle, subtract this from the desired parameter of dff2phase organelle to obtain the base delay, interpolate invl<sub>x</sub> for the desired fanout plus one and add this to the base delay to get the desired delay times. Hold times, setup times, clock1 to T3, and data to T3 are independent of fanout.

Figure A.6b Dff2phase Timing Data

---

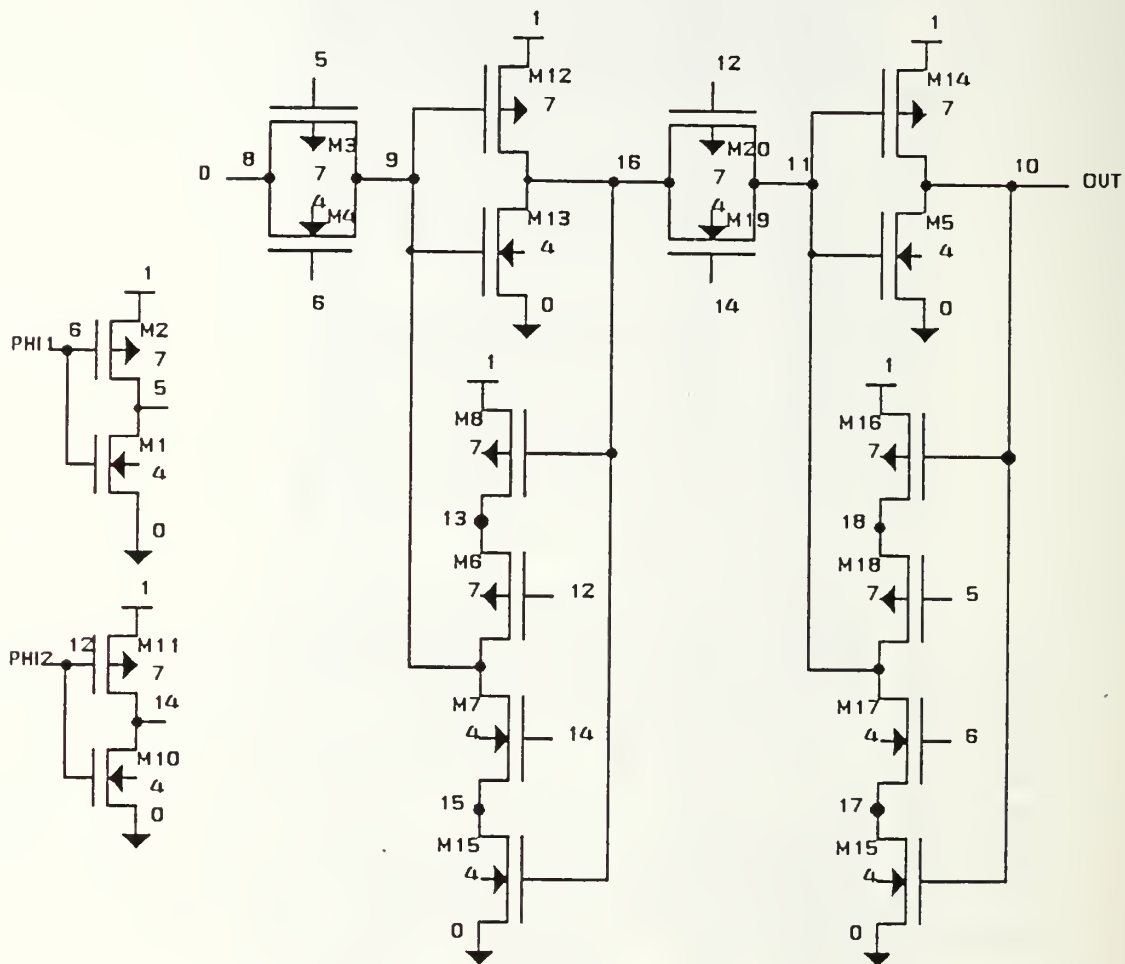


Figure A.6c Dff2phase Schematic



cifplot\* Window: -3150 2100 -1650 2100 --- Scale: 1 micron is 0.1 inches (2540x)

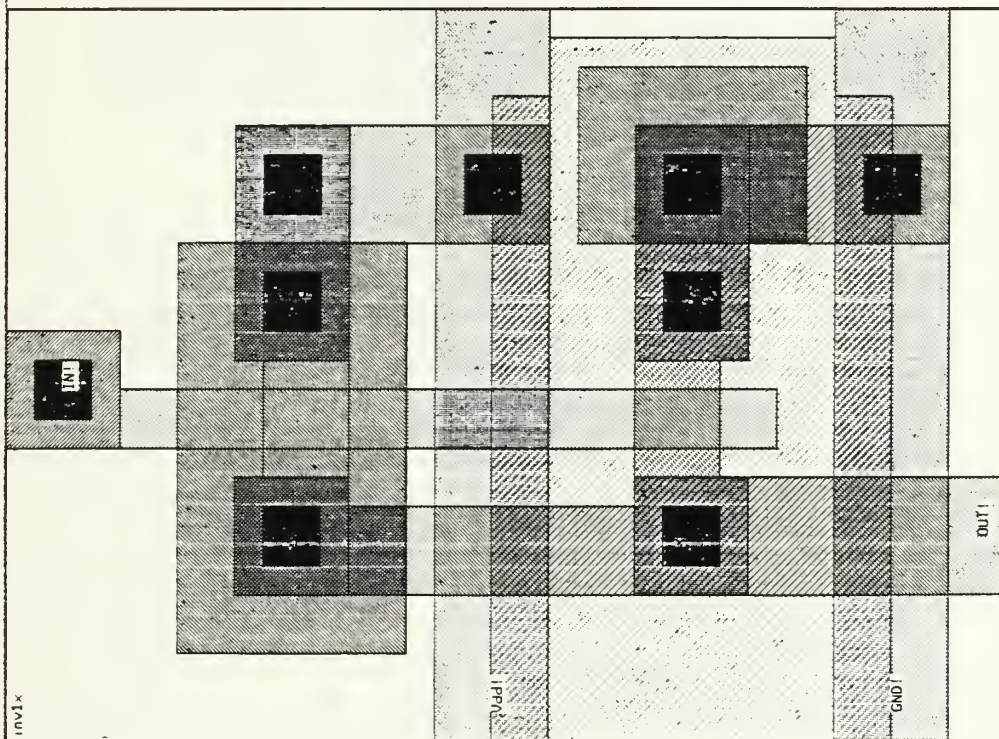


Figure A.7a Inv1x Cif Plot

---

| FAN OUT | $t_{dr}$ | $t_{df}$ | $t_r$  | $t_f$ |
|---------|----------|----------|--------|-------|
| 1       | 0.75ns   | 0.6ns    | 1.35ns | 0.9ns |
| 4       | 1.5ns    | 0.95ns   | 3.4ns  | 1.7ns |

| TRUTH TABLE |     |
|-------------|-----|
| IN          | OUT |
| 0           | 1   |
| 1           | 0   |

Figure A.7b Inv1x Timing Data

---

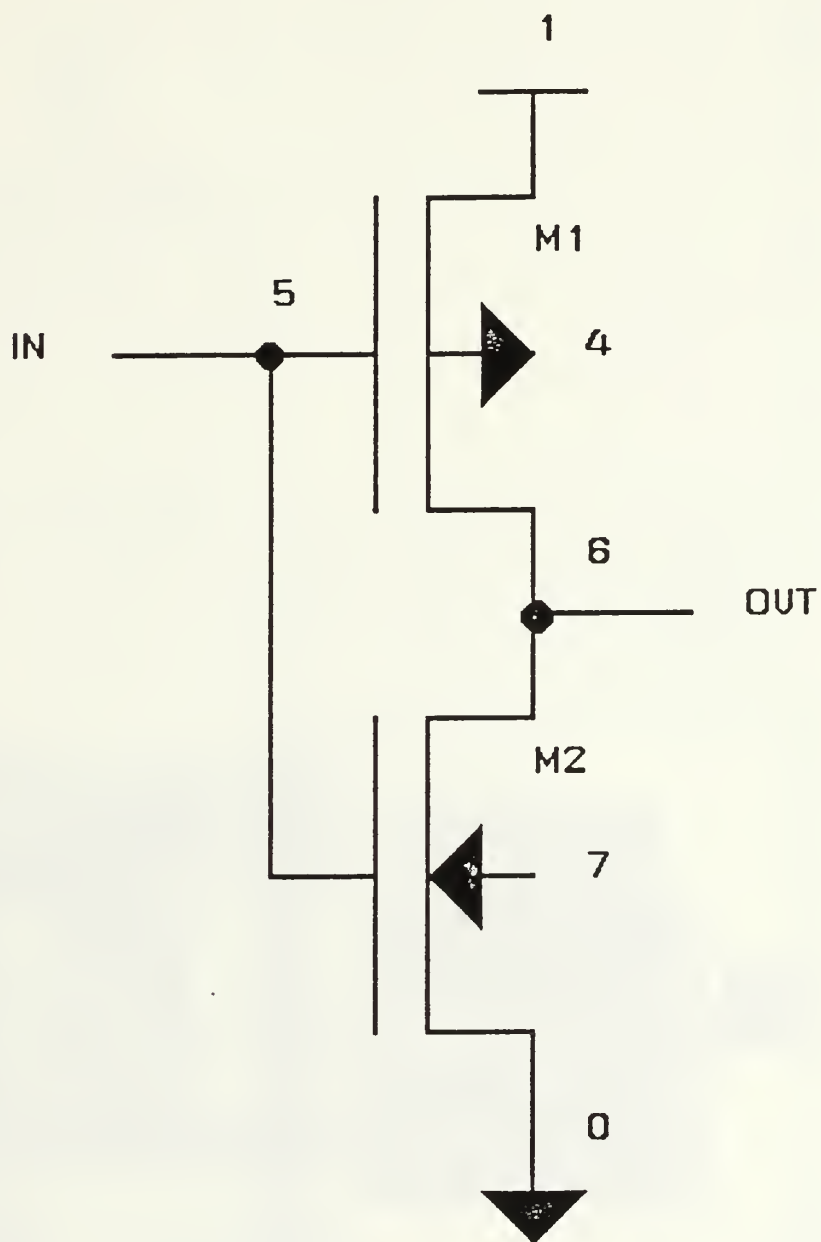


Figure A.7c Inv1x Schematic

cifplot\* Win. ow: -1650 4800 -1350 2400 --- Scale: 1 micron is 0.075 inches (1905x)

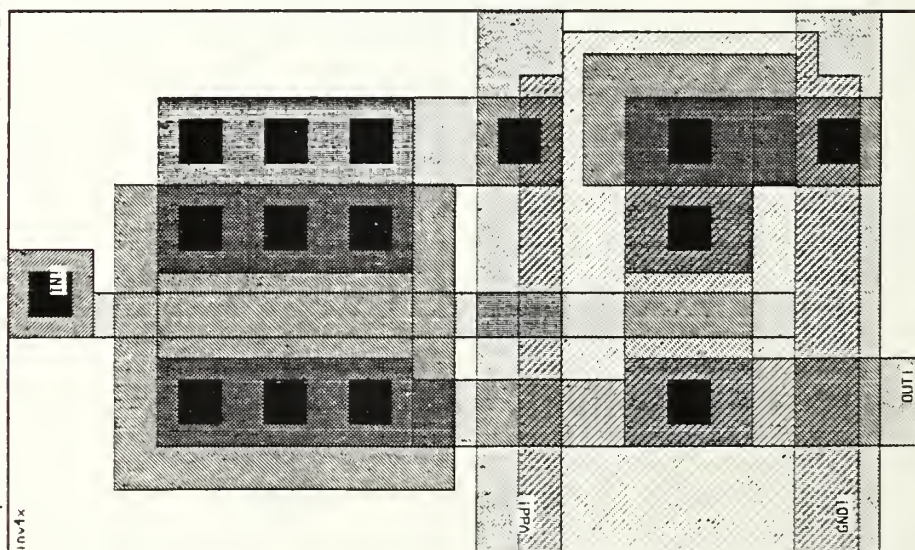


Figure A.8a Inv4x Cif Plot

---

| FAN OUT | $t_{dr}$ | $t_{df}$ | $t_r$  | $t_f$  |
|---------|----------|----------|--------|--------|
| 0.25    | 0.45ns   | 0.7ns    | 0.8ns  | 1.15ns |
| 1       | 0.7ns    | 1.1ns    | 1.35ns | 1.55ns |
| 4       | 1.9ns    | 2.0ns    | 3.75ns | 3.1ns  |

| TRUTH TABLE |     |
|-------------|-----|
| IN          | OUT |
| 0           | 1   |
| 1           | 0   |

Figure A.8b Inv4x Timing Data

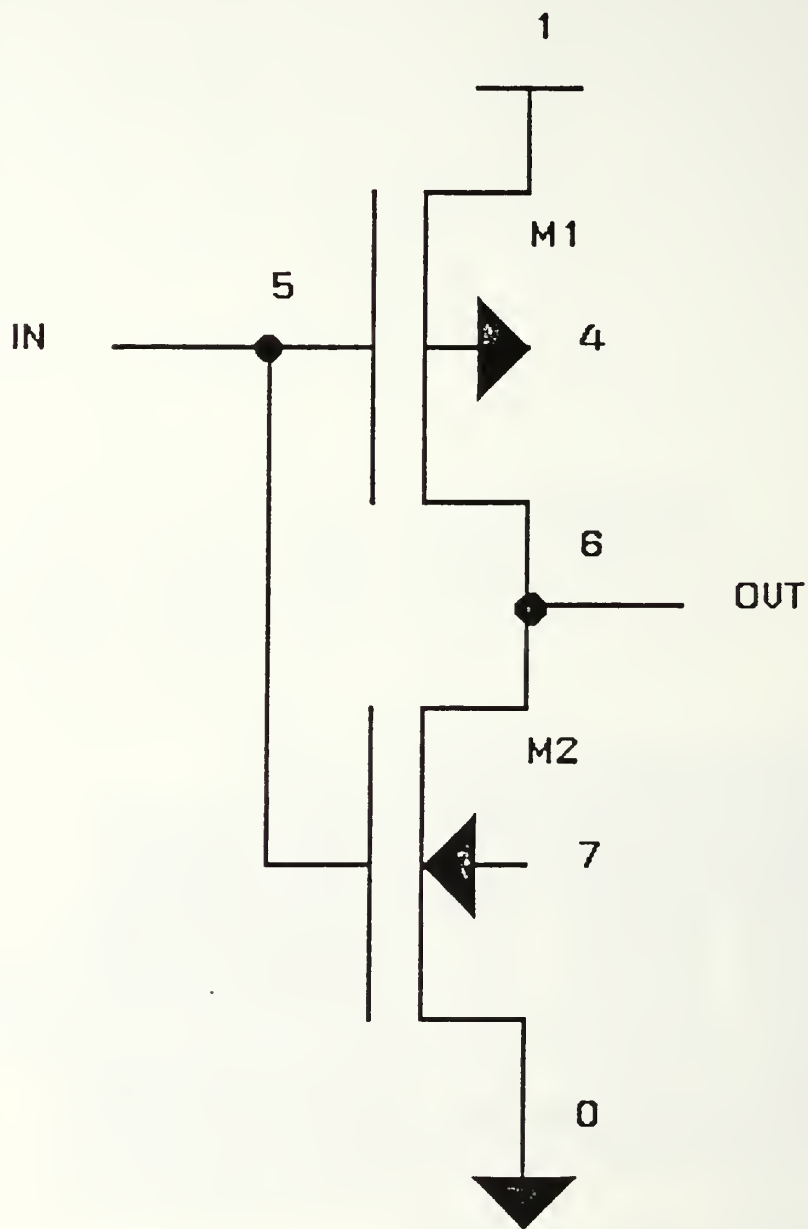


Figure A.8c Inv4x Schematic



cifplot\* Window: -5550 3600 -1200 2550 --- Scale: 1 micron is 0.075 inches

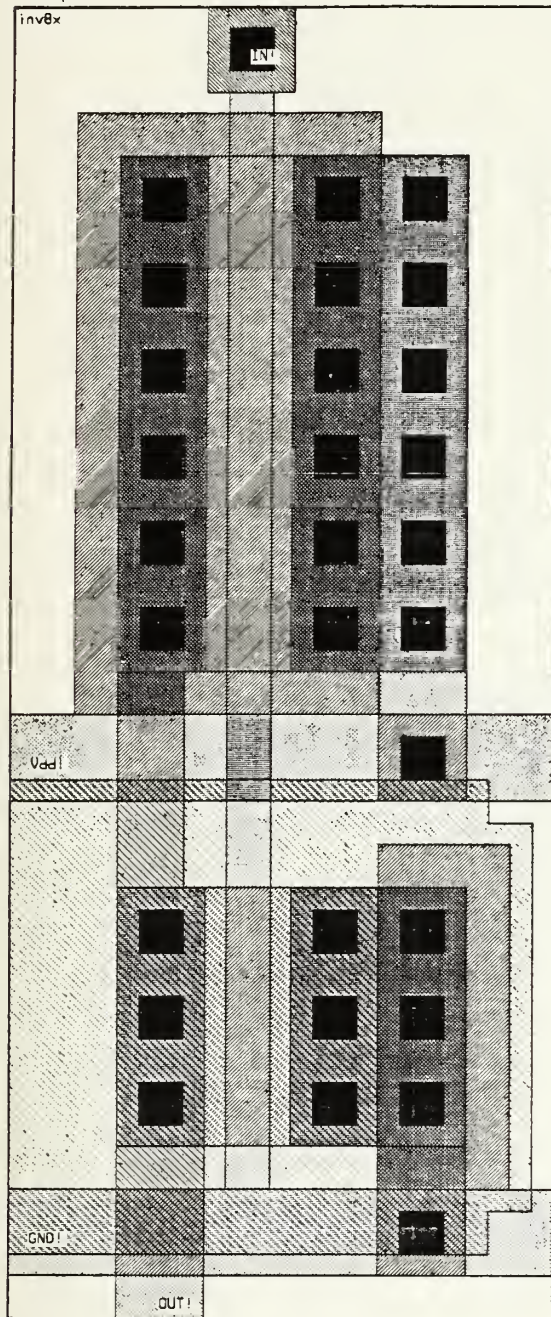


Figure A.9a Inv8x Cif Plot

---

| FAN OUT | $t_{dr}$ | $t_{df}$ | $t_r$  | $t_f$ |
|---------|----------|----------|--------|-------|
| 1       | 0.8ns    | 1.0ns    | 1.6ns  | 1.6ns |
| 4       | 1.9ns    | 2.0ns    | 3.75ns | 3.1ns |

| TRUTH TABLE |     |
|-------------|-----|
| IN          | OUT |
| 0           | 1   |
| 1           | 0   |

Figure A.9b Inv8x Timing Data

---

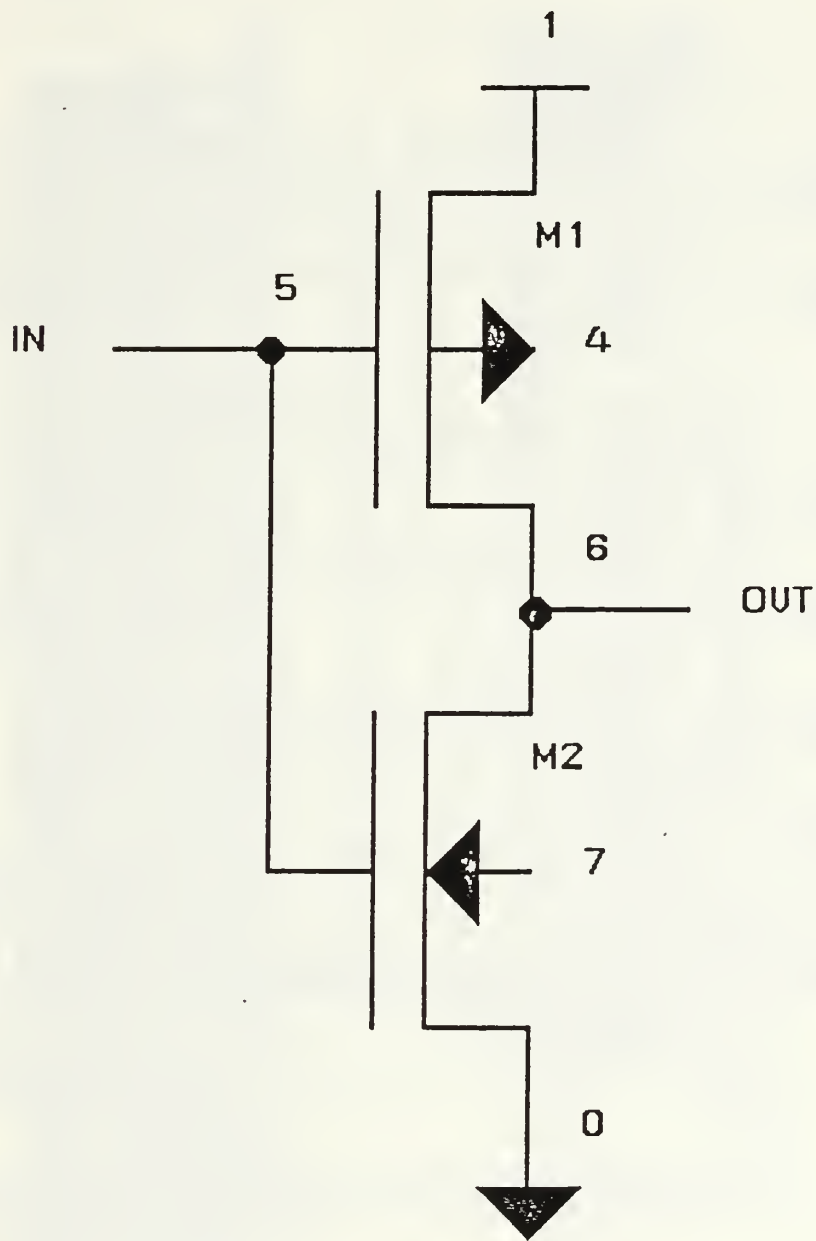


Figure A.9c Inv8x Schematic



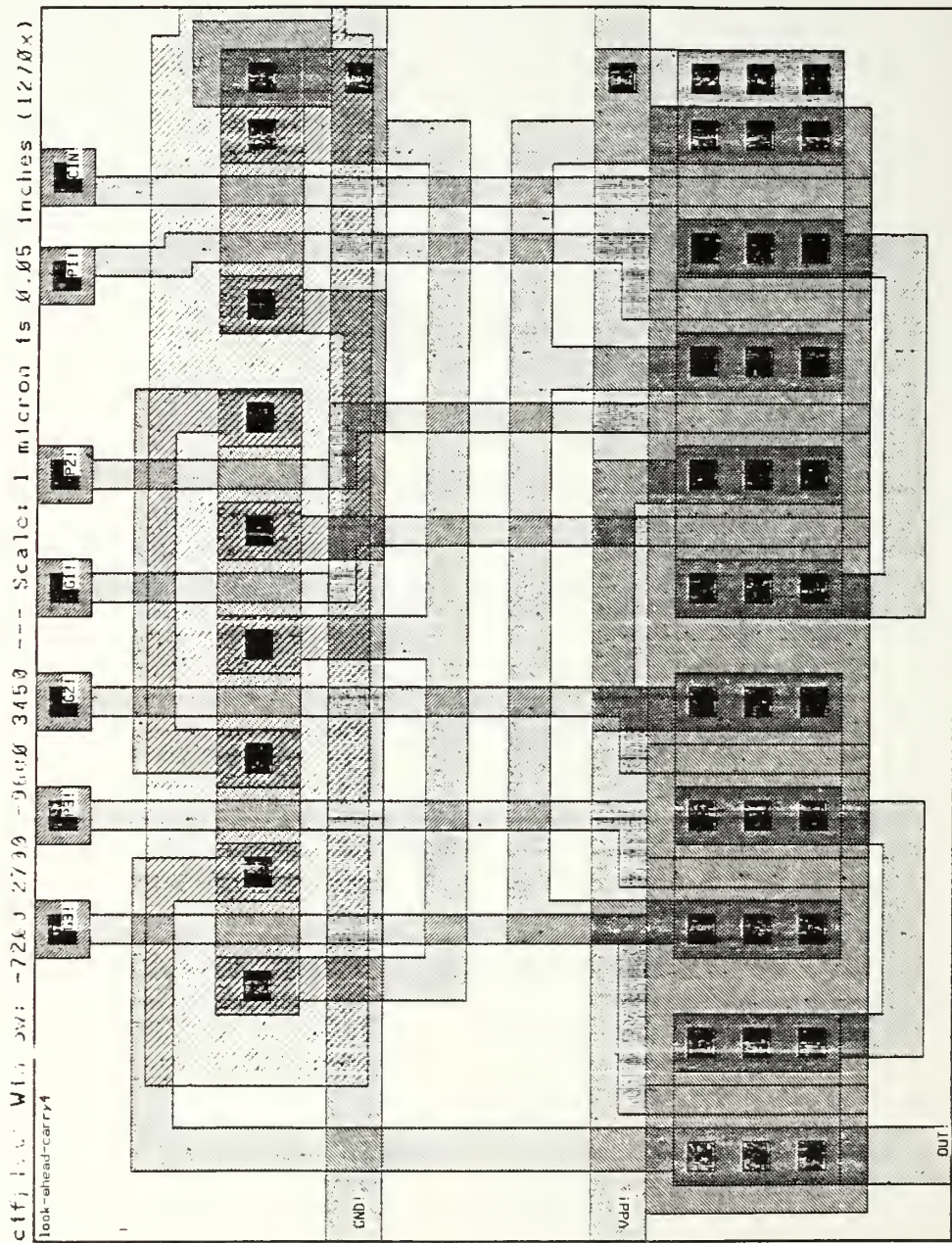


Figure A.10a Look-ahead-carry4 Cif Plot

| INPUT | FAN OUT | $t_{dr}$ | $t_{df}$ | $t_r$ | $t_f$ |
|-------|---------|----------|----------|-------|-------|
| CIN1  | 3.6ns   | 2.2ns    | 4.8ns    | 4.6ns |       |
|       | 4       | 4.1ns    | 2.5ns    | 6.5ns | 6.3ns |
| P1    | 1       | 2.8ns    | 2.6ns    | 4.6ns | 4.6ns |
|       | 4       | 3.4ns    | 3.0ns    | 6.3ns | 6.3ns |
| P2    | 1       | 1.9ns    | 2.8ns    | 3.1ns | 4.6ns |
|       | 4       | 2.3ns    | 3.3ns    | 4.7ns | 6.3ns |
| P3    | 1       | 1.1ns    | 2.6ns    | 2.5ns | 4.6ns |
|       | 4       | 1.2ns    | 3.4ns    | 3.6ns | 6.3ns |
| G1    | 1       | 2.8ns    | 2.6ns    | 4.6ns | 4.6ns |
|       | 4       | 3.4ns    | 3.0ns    | 6.3ns | 6.3ns |
| G2    | 1       | 1.9ns    | 2.8ns    | 3.1ns | 4.6ns |
|       | 4       | 2.3ns    | 3.3ns    | 4.7ns | 6.3ns |
| G3    | 1       | 1.1ns    | 2.6ns    | 2.5ns | 4.6ns |
|       | 4       | 1.2ns    | 3.4ns    | 3.6ns | 6.3ns |

| TRUTH TABLE      |    |    |     |    |    |    |     |
|------------------|----|----|-----|----|----|----|-----|
| P3               | P2 | P1 | CIN | G3 | G2 | G1 | OUT |
| 1                | 1  | 1  | 1   | X  | X  | X  | 0   |
| 1                | 1  | X  | X   | X  | X  | 1  | 0   |
| 1                | X  | X  | X   | X  | 1  | X  | 0   |
| X                | X  | X  | X   | 1  | X  | X  | 0   |
| ALL OTHER INPUTS |    |    |     |    |    |    | 1   |

- This is stage three of a four stage static look ahead carry, where:  
 $OUT = G3 + P3(G2 + P2(G1 + P1 \cdot CIN))$
- Stage four is obtained by:  $OUT = G4 + P4 \cdot OUT_{STAGE}$
- Stage two is obtained by setting  $G3 = 0$  and  $P3 = 1$  in stage three. Stage one is obtained by using individual organelles to generate:  
 $OUT = G1 + P1 \cdot CIN$

Figure A.10b Look-ahead-carry4 Timing Data

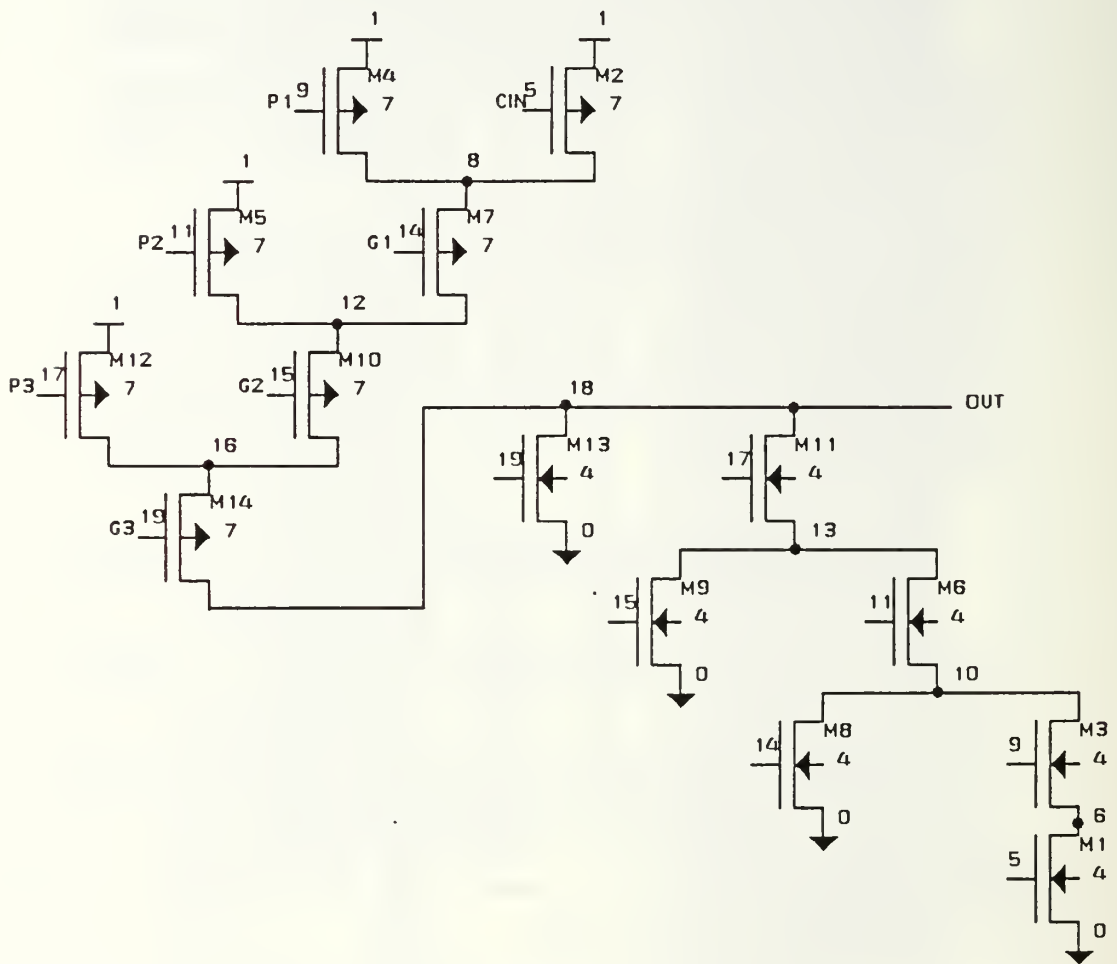


Figure A.10c Look-ahead-carry4 Schematic



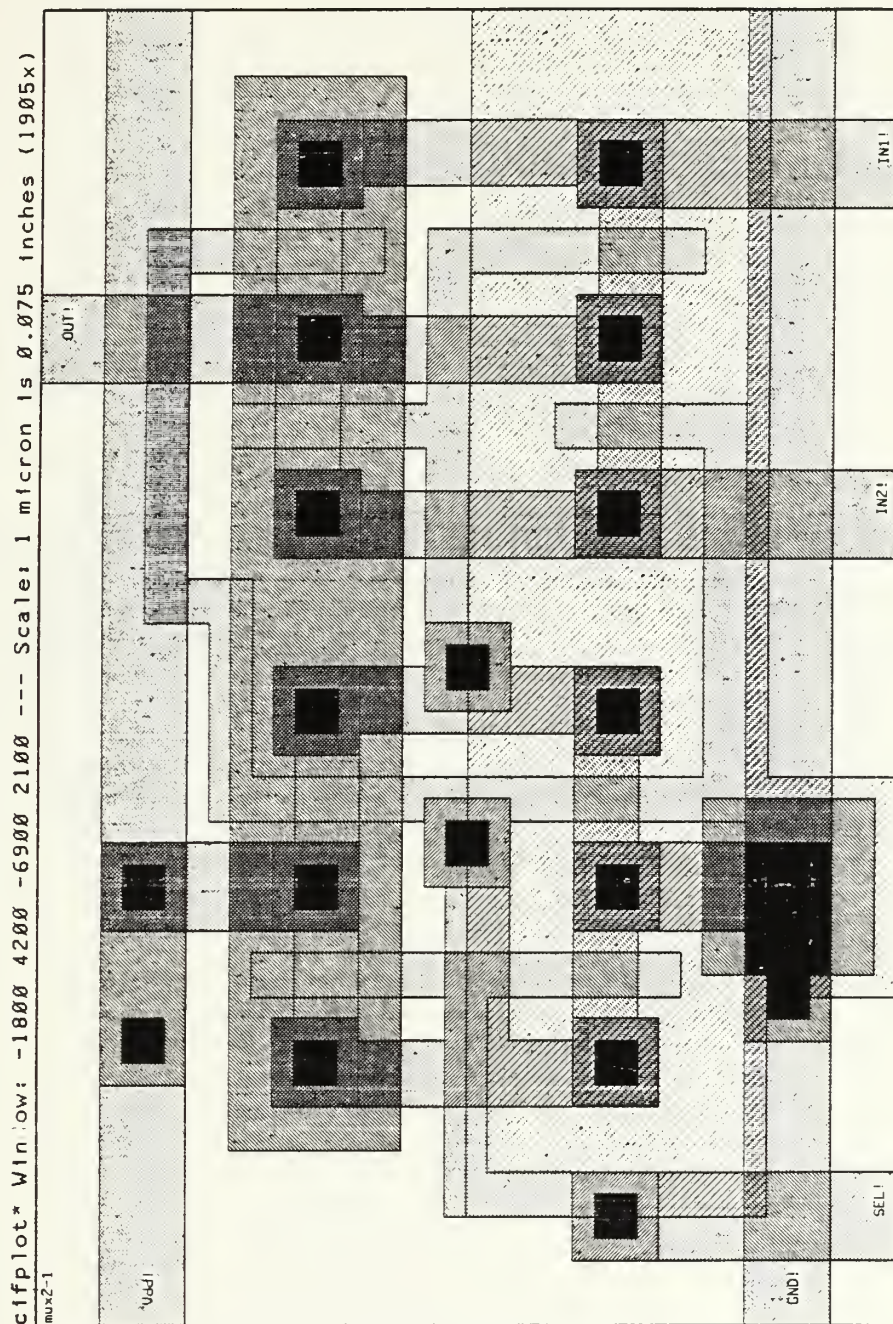


Figure A.11a Mux2-1 Cif Plot

---

| FANOUT | $t_{dr}$ | $t_{df}$ | $t_r$ | $t_f$ | SEL TO OUT |
|--------|----------|----------|-------|-------|------------|
| 1      | 0.6ns    | 0.4ns    | 3.4ns | 1.5ns | 3.0ns      |
| 4      | 1.2ns    | 6.7ns    | 0.9ns | 2.7ns | 3.6ns      |

| TRUTH TABLE |     |
|-------------|-----|
| SEL         | OUT |
| 0           | IN2 |
| 1           | IN1 |

Figure A.11b Mux2-1 Timing Data

---

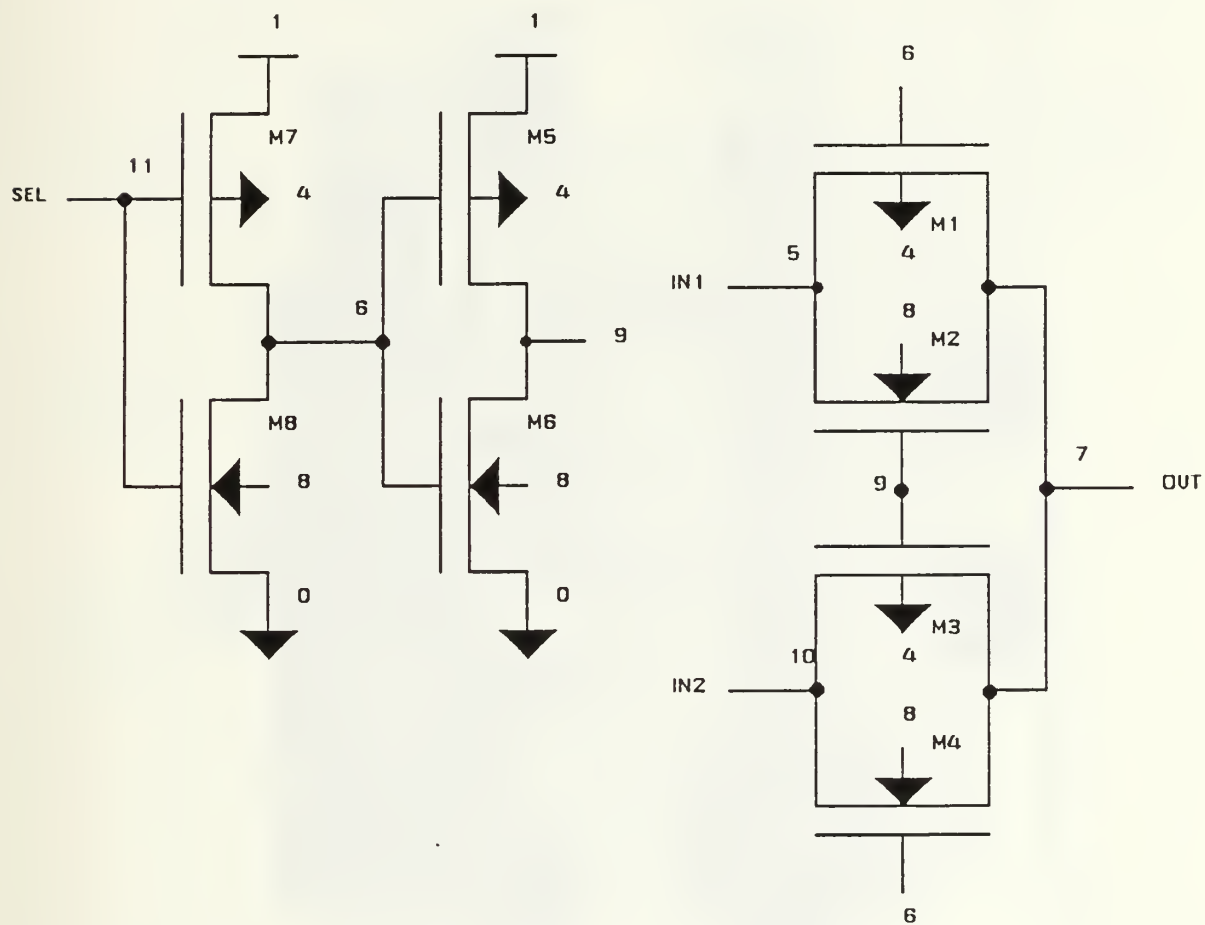


Figure A.11c Mux2-1 Schematic



cifplot\* Window: -4200 300 -2850 2700 --- Scale: 1 micron is 0.1 inches (2540x)

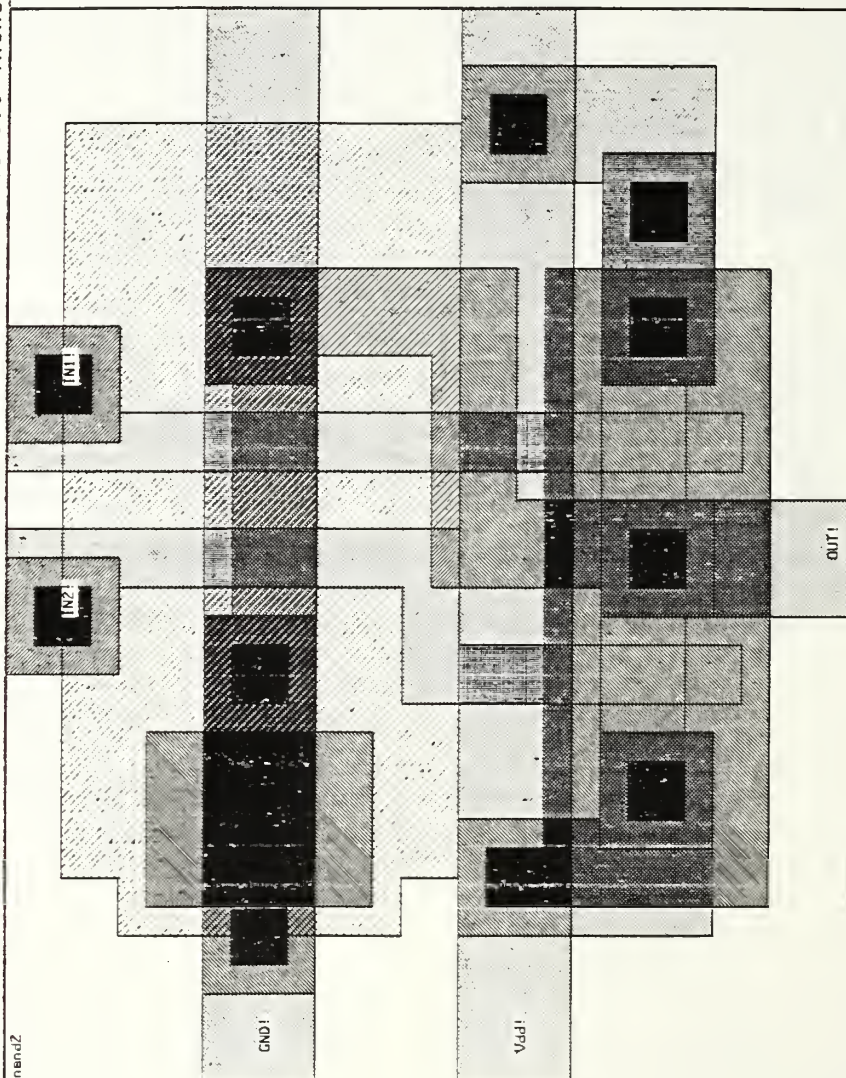


Figure A.12a Nand2 Cif Plot

---

| INPUT | FANOUT | $t_{dr}$ | $t_{df}$ | $t_r$ | $t_f$ |
|-------|--------|----------|----------|-------|-------|
| IN1   | 1      | 0.8ns    | 0.8ns    | 1.5ns | 1.4ns |
|       | 4      | 1.7ns    | 1.5ns    | 3.5ns | 2.9ns |
| IN2   | 1      | 1.0ns    | 0.9ns    | 1.7ns | 1.4ns |
|       | 4      | 1.8ns    | 1.5ns    | 3.7ns | 2.9ns |

| TRUTH TABLE |     |     |
|-------------|-----|-----|
| IN1         | IN2 | OUT |
| 0           | 0   | 1   |
| 1           | 0   | 1   |
| 0           | 1   | 1   |
| 1           | 1   | 0   |

Figure A.12b Nand2 Timing Data

---

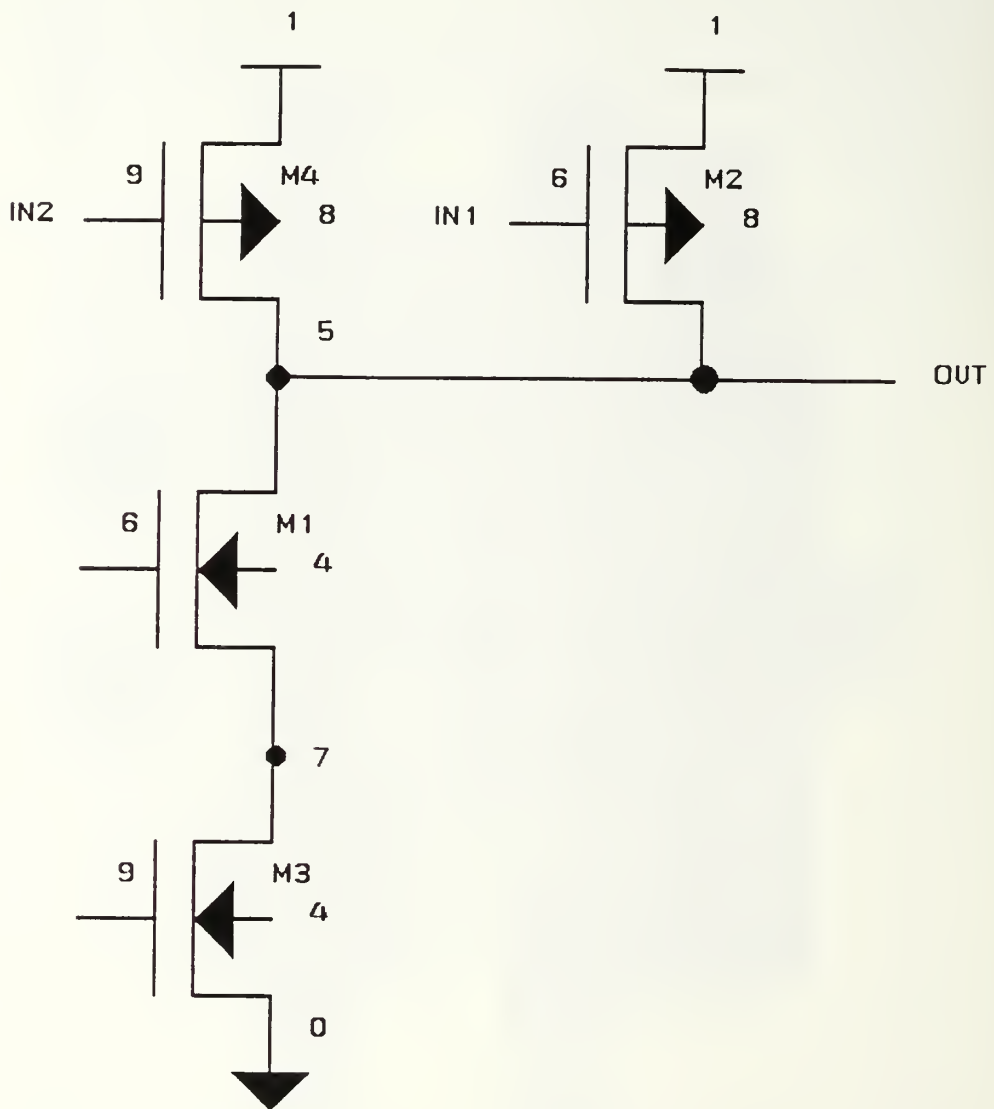


Figure A.12c NAND2 Schematic



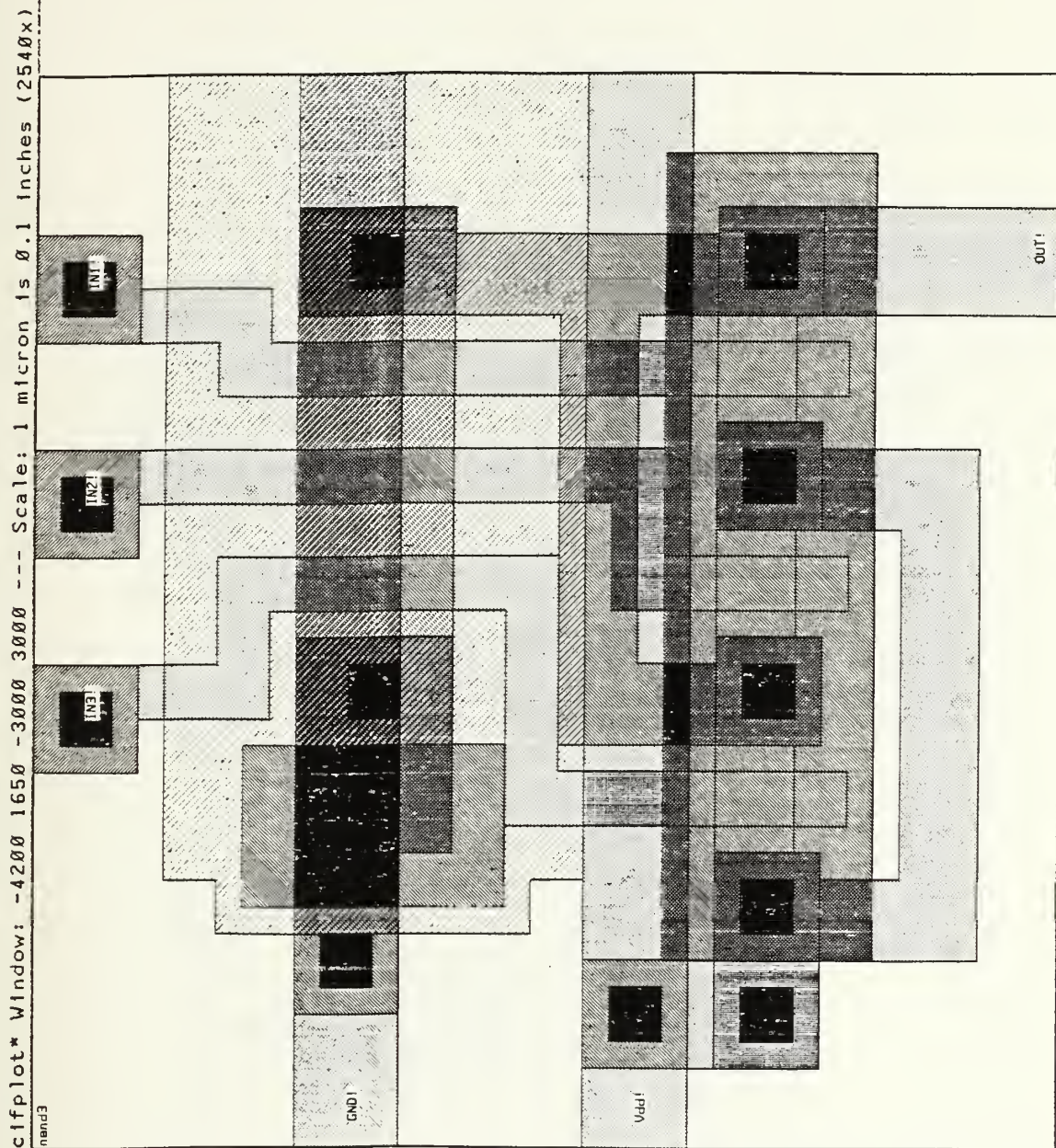


Figure A.13a Nand3 Cif Plot

---

| INPUT | FANOUT | $t_{dr}$ | $t_{df}$ | $t_r$ | $t_f$ |
|-------|--------|----------|----------|-------|-------|
| IN1   | 1      | 1.2ns    | 0.8ns    | 1.8ns | 1.3ns |
|       | 4      | 2.0ns    | 1.2ns    | 3.7ns | 2.4ns |
| IN2   | 1      | 1.5ns    | 0.8ns    | 2.1ns | 1.3ns |
|       | 4      | 2.4ns    | 1.2ns    | 3.9ns | 2.4ns |
| IN3   | 1      | 1.8ns    | 0.8ns    | 2.6ns | 1.3ns |
|       | 4      | 2.6ns    | 1.2ns    | 4.4ns | 2.4ns |

| TRUTH TABLE |     |     |     |
|-------------|-----|-----|-----|
| IN1         | IN2 | IN3 | OUT |
| 0           | 0   | 0   | 1   |
| 1           | 0   | 0   | 1   |
| 0           | 1   | 0   | 1   |
| 1           | 1   | 0   | 1   |
| 0           | 0   | 1   | 1   |
| 1           | 0   | 1   | 1   |
| 0           | 1   | 1   | 1   |
| 1           | 1   | 1   | 0   |

Figure A.13b Nand3 Timing Data

---

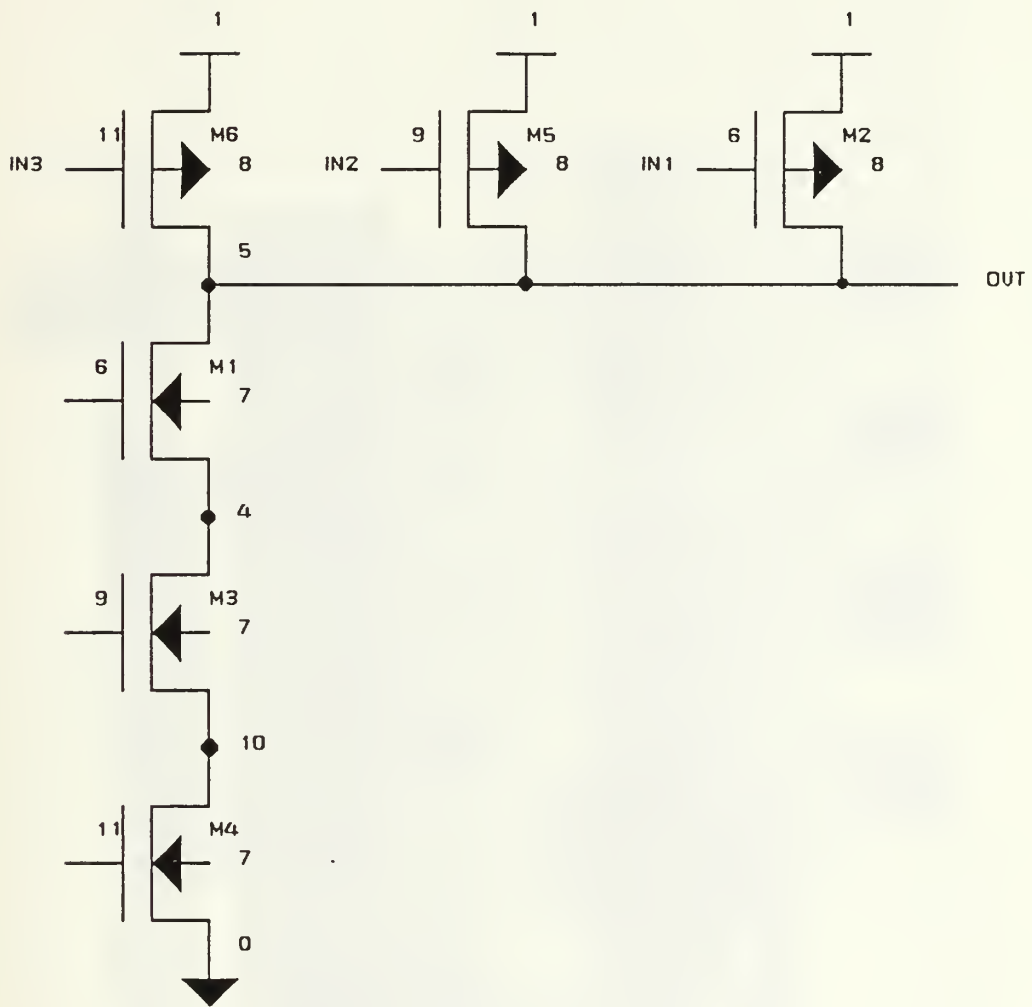


Figure A.13c Nand3 Schematic



cifplot\* Window: -3900 2550 -3150 3600 --- Scale: 1 micron is 0.075 inches (1905x)

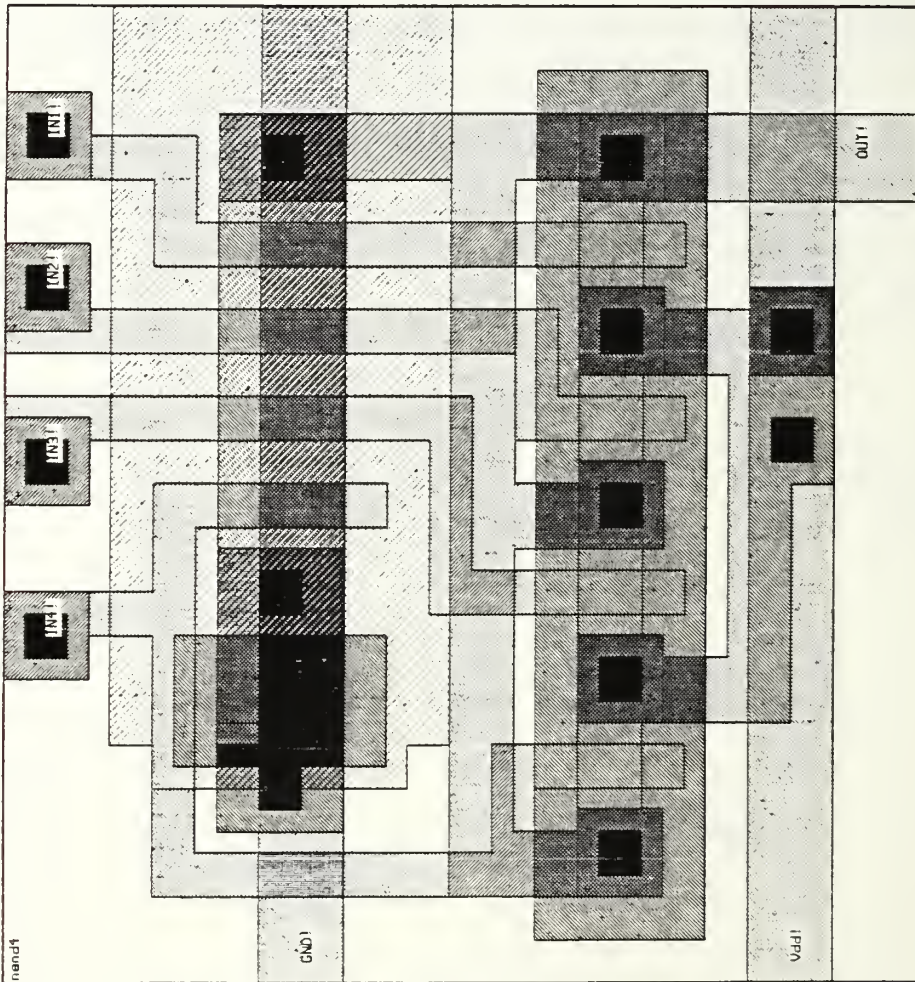


Figure A.14a Nand4 Cif Plot

---

| INPUT | FANOUT | $t_{dr}$ | $t_{df}$ | $t_r$ | $t_f$ |
|-------|--------|----------|----------|-------|-------|
| IN1   | 1      | 1.2ns    | 1.0ns    | 1.8ns | 1.9ns |
|       | 4      | 2.0ns    | 1.5ns    | 3.9ns | 3.3ns |
| IN2   | 1      | 1.6ns    | 1.0ns    | 2.2ns | 1.9ns |
|       | 4      | 2.4ns    | 1.6ns    | 4.0ns | 3.3ns |
| IN3   | 1      | 1.8ns    | 1.1ns    | 2.5ns | 1.9ns |
|       | 4      | 2.7ns    | 1.6ns    | 4.1ns | 3.3ns |
| IN4   | 1      | 2.1ns    | 1.1ns    | 3.1ns | 1.9ns |
|       | 4      | 3.0ns    | 1.6ns    | 3.9ns | 3.4ns |

| TRUTH TABLE |     |     |     |     |
|-------------|-----|-----|-----|-----|
| IN1         | IN2 | IN3 | IN4 | OUT |
| 0           | 0   | 0   | 0   | 1   |
| 1           | 0   | 0   | 0   | 1   |
| 0           | 1   | 0   | 0   | 1   |
| 1           | 1   | 0   | 0   | 1   |
| 0           | 0   | 1   | 0   | 1   |
| 1           | 0   | 1   | 0   | 1   |
| 0           | 1   | 1   | 0   | 1   |
| 1           | 1   | 1   | 0   | 1   |
| 0           | 0   | 0   | 1   | 1   |
| 1           | 0   | 0   | 1   | 1   |
| 0           | 1   | 0   | 1   | 1   |
| 1           | 1   | 0   | 1   | 1   |
| 0           | 0   | 1   | 1   | 1   |
| 1           | 0   | 1   | 1   | 1   |
| 0           | 1   | 1   | 1   | 1   |
| 1           | 1   | 1   | 1   | 0   |

Figure A.14b Nand4 Timing Data

---

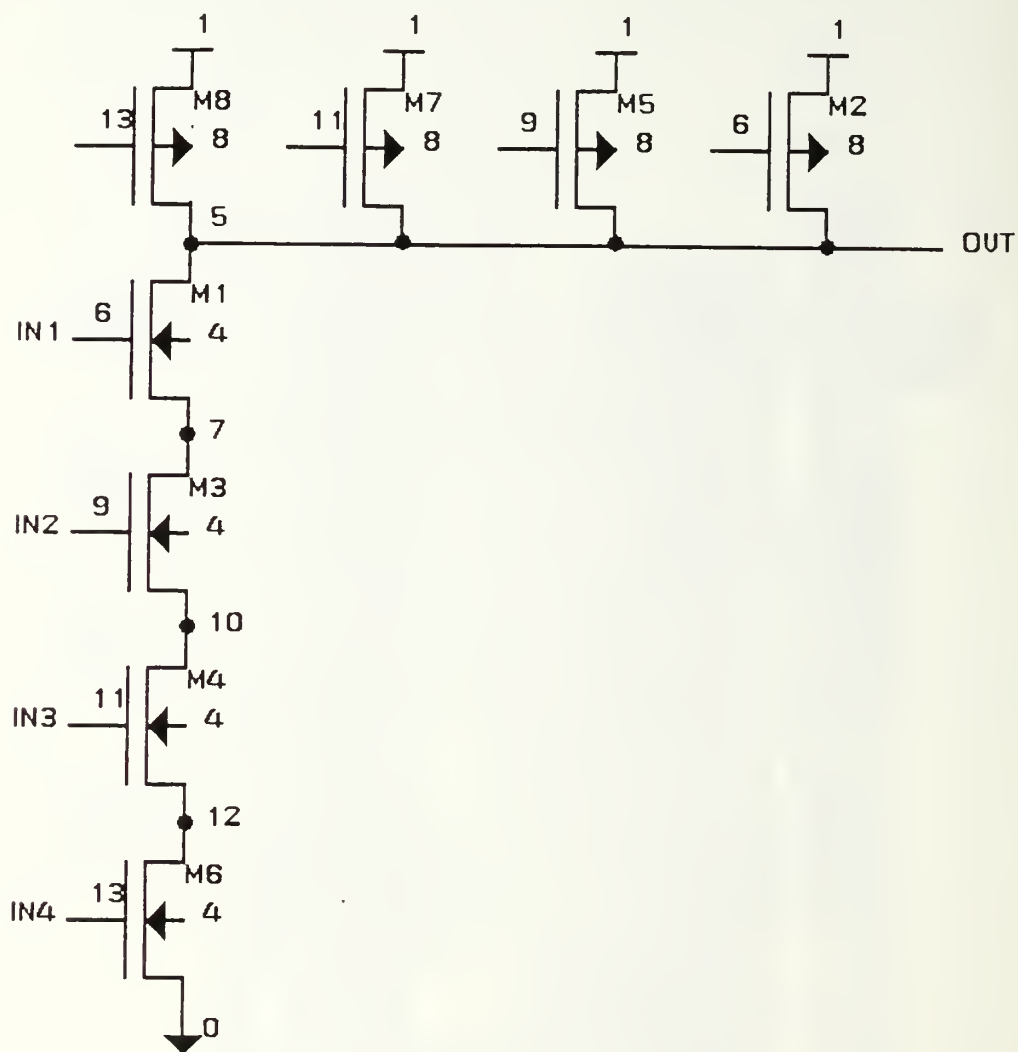


Figure A.14c Nand4 Schematic



cifplot\* Window: -4500 300 -2700 2700 --- Scale: 1 micron is 0.1 inches (2540x)

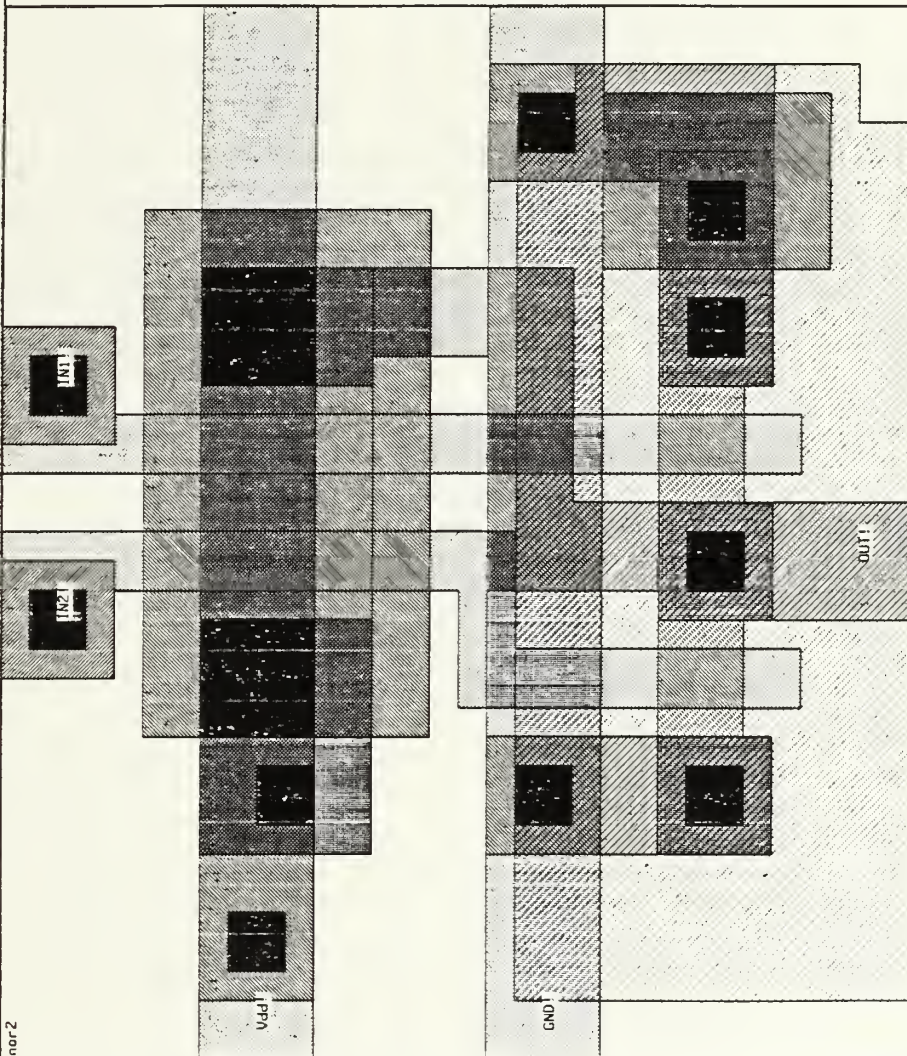


Figure A.15a Nor2 Cif Plot

---

| INPUT | FANOUT | $t_{dr}$ | $t_{df}$ | $t_r$ | $t_f$ |
|-------|--------|----------|----------|-------|-------|
| IN1   | 1      | 0.9ns    | 0.8ns    | 1.9ns | 1.1ns |
|       | 4      | 1.8ns    | 1.2ns    | 3.9ns | 1.8ns |
| IN2   | 1      | 1.0ns    | 0.8ns    | 1.9ns | 1.3ns |
|       | 4      | 1.9ns    | 1.2ns    | 3.9ns | 2.0ns |

| TRUTH TABLE |     |     |
|-------------|-----|-----|
| IN1         | IN2 | OUT |
| 0           | 0   | 1   |
| 1           | 0   | 0   |
| 0           | 1   | 0   |
| 1           | 1   | 0   |

Figure A.15b Nor2 Timing Data

---

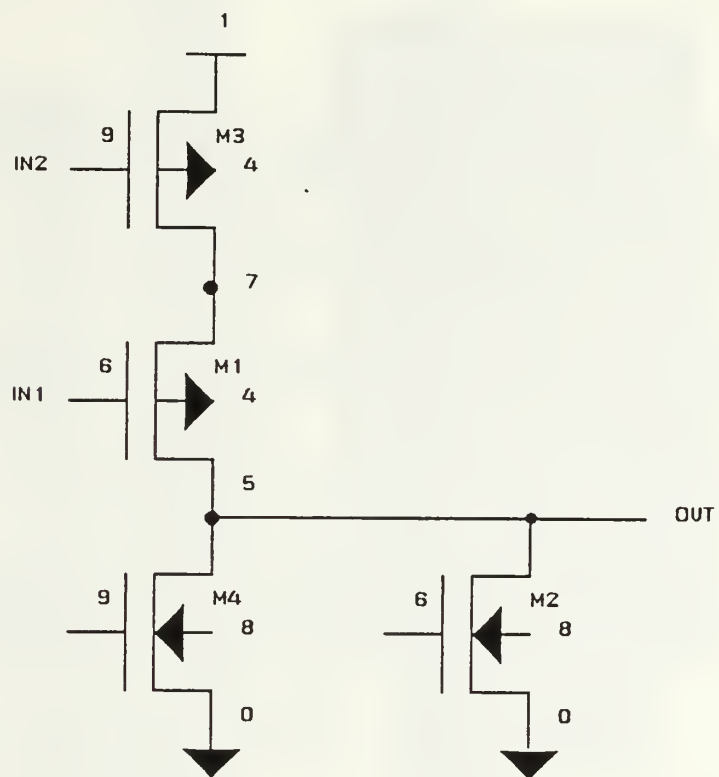


Figure A.15c Nor2 Schematic



cifplot\* Window: -4650 1500 -3150 3000 --- Scale: 1 micron is 0.1 inches (2540x)

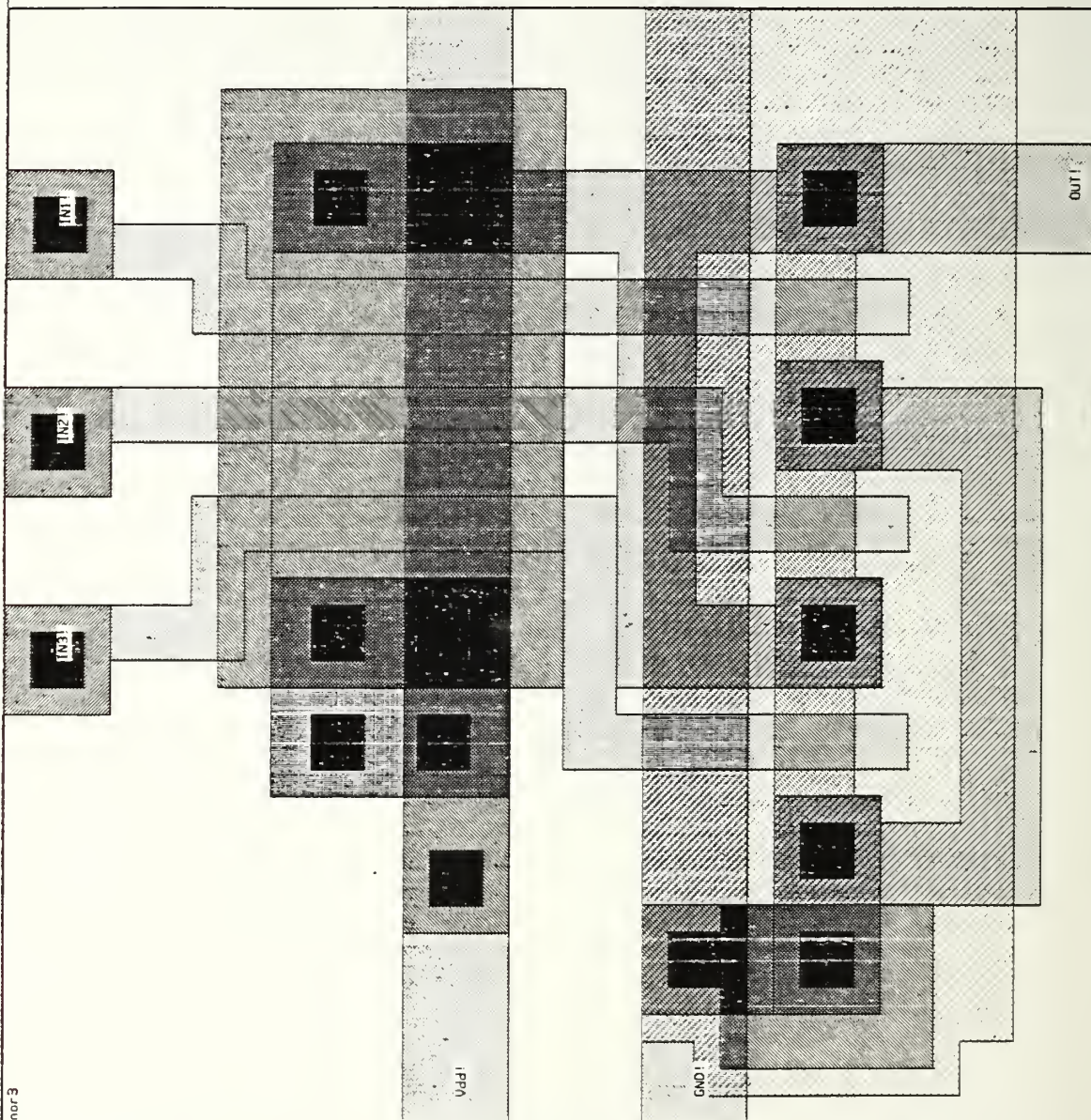


Figure A.16a Nor3 Cif Plot

---

| INPUT | FANOUT | $t_{dr}$ | $t_{df}$ | $t_r$ | $t_f$ |
|-------|--------|----------|----------|-------|-------|
| IN1   | 1      | 1.0ns    | 1.0ns    | 2.6ns | 1.2ns |
|       | 4      | 1.9ns    | 1.5ns    | 4.5ns | 2.1ns |
| IN2   | 1      | 1.4ns    | 1.2ns    | 2.6ns | 1.7ns |
|       | 4      | 2.3ns    | 1.6ns    | 4.5ns | 2.3ns |
| IN3   | 1      | 1.5ns    | 1.2ns    | 2.7ns | 1.9ns |
|       | 4      | 2.4ns    | 1.6ns    | 4.5ns | 2.3ns |

| TRUTH TABLE |     |     |     |
|-------------|-----|-----|-----|
| IN1         | IN2 | IN3 | OUT |
| 0           | 0   | 0   | 1   |
| 1           | 0   | 0   | 0   |
| 0           | 1   | 0   | 0   |
| 1           | 1   | 0   | 0   |
| 0           | 0   | 1   | 0   |
| 1           | 0   | 1   | 0   |
| 0           | 1   | 1   | 0   |
| 1           | 1   | 1   | 0   |

Figure A.16b Nor3 Timing Data

---

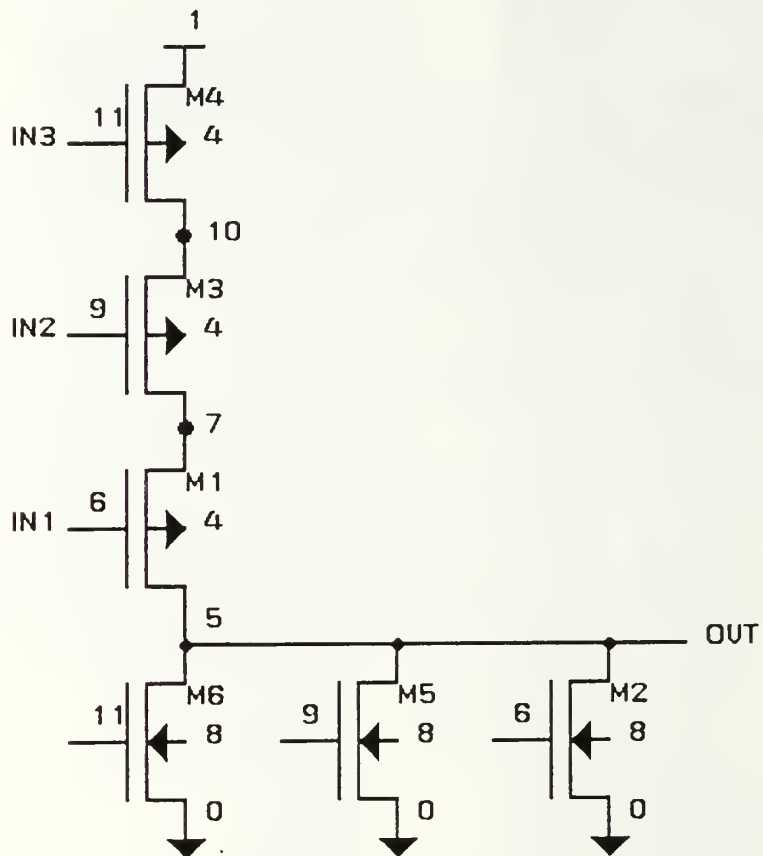


Figure A.16c Nor3 Schematic



cifplot\* Window: -4800 2400 -3300 3600 --- Scale: 1 micron is 0.075 inches (1905x)

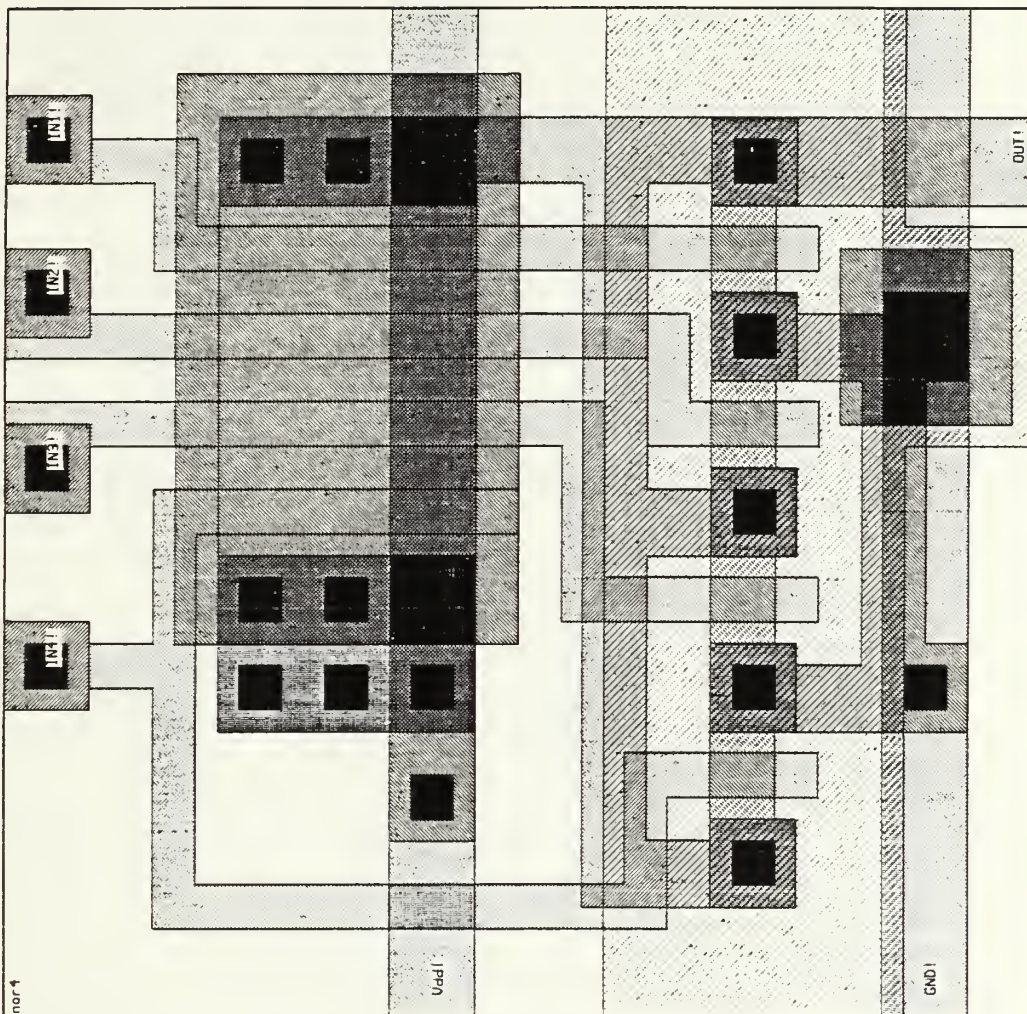


Figure A.17a Nor4 Cif Plot

---

| INPUT | FANOUT | $t_{dr}$ | $t_{df}$ | $t_r$ | $t_f$ |
|-------|--------|----------|----------|-------|-------|
| IN1   | 1      | 1.1ns    | 1.4ns    | 3.3ns | 1.4ns |
|       | 4      | 1.9ns    | 1.8ns    | 5.3ns | 2.0ns |
| IN2   | 1      | 1.7ns    | 1.6ns    | 3.5ns | 1.8ns |
|       | 4      | 2.6ns    | 1.9ns    | 5.7ns | 2.5ns |
| IN3   | 1      | 2.0ns    | 1.7ns    | 3.6ns | 2.4ns |
|       | 4      | 2.8ns    | 2.0ns    | 5.9ns | 3.1ns |
| IN4   | 1      | 2.0ns    | 1.7ns    | 3.8ns | 2.7ns |
|       | 4      | 2.9ns    | 2.1ns    | 6.1ns | 4.9ns |

| TRUTH TABLE |     |     |     |     |
|-------------|-----|-----|-----|-----|
| IN1         | IN2 | IN3 | IN4 | OUT |
| 0           | 0   | 0   | 0   | 1   |
| 1           | 0   | 0   | 0   | 0   |
| 0           | 1   | 0   | 0   | 0   |
| 1           | 1   | 0   | 0   | 0   |
| 0           | 0   | 1   | 0   | 0   |
| 1           | 0   | 1   | 0   | 0   |
| 0           | 1   | 1   | 0   | 0   |
| 1           | 1   | 1   | 0   | 0   |
| 0           | 0   | 0   | 1   | 0   |
| 1           | 0   | 0   | 1   | 0   |
| 0           | 1   | 0   | 1   | 0   |
| 1           | 1   | 0   | 1   | 0   |
| 0           | 0   | 1   | 1   | 0   |
| 1           | 0   | 1   | 1   | 0   |
| 0           | 1   | 1   | 1   | 0   |
| 1           | 1   | 1   | 1   | 0   |

Figure A.17b Nor4 Timing Data

---

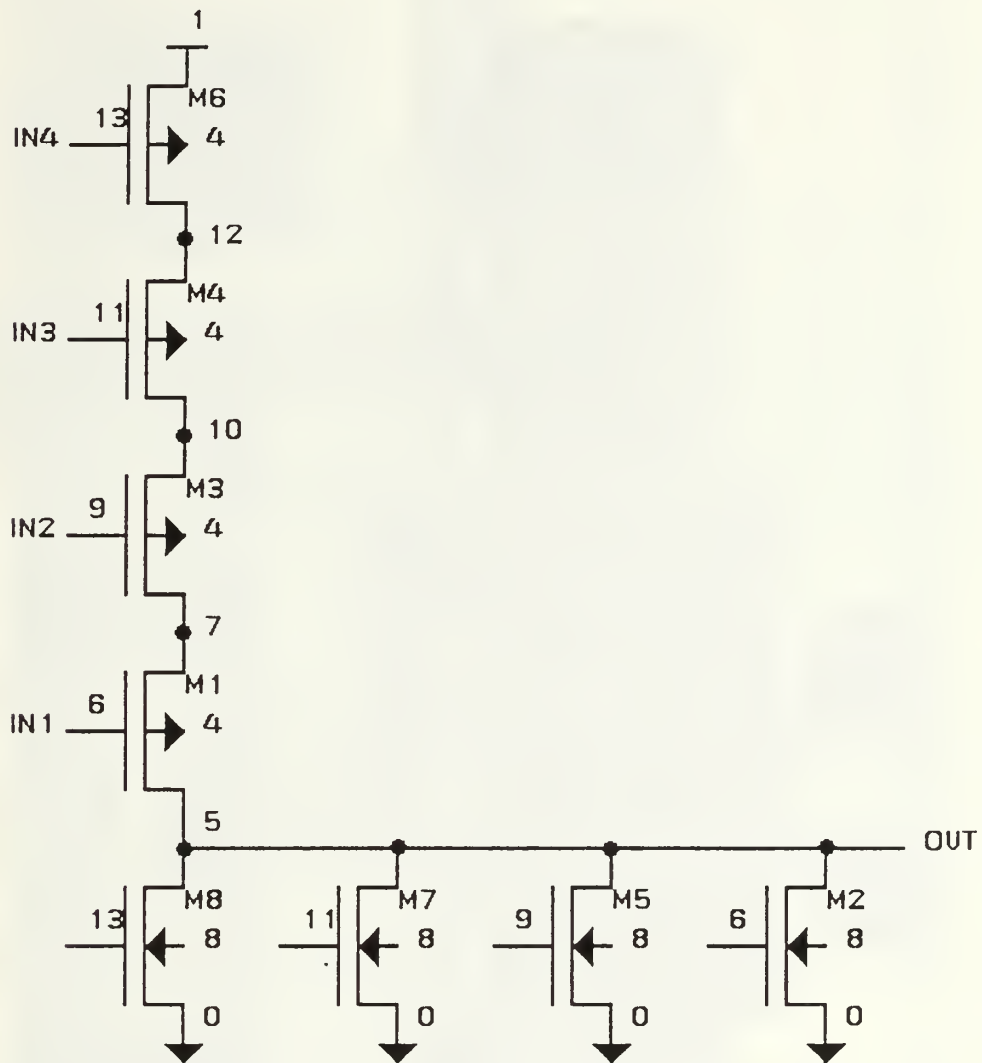


Figure A.17c Nor4 Schematic



cifplot\* Window: -4500 600 -2400 4500 --- Scale: 1 micron is 0.1 inches (2540x)

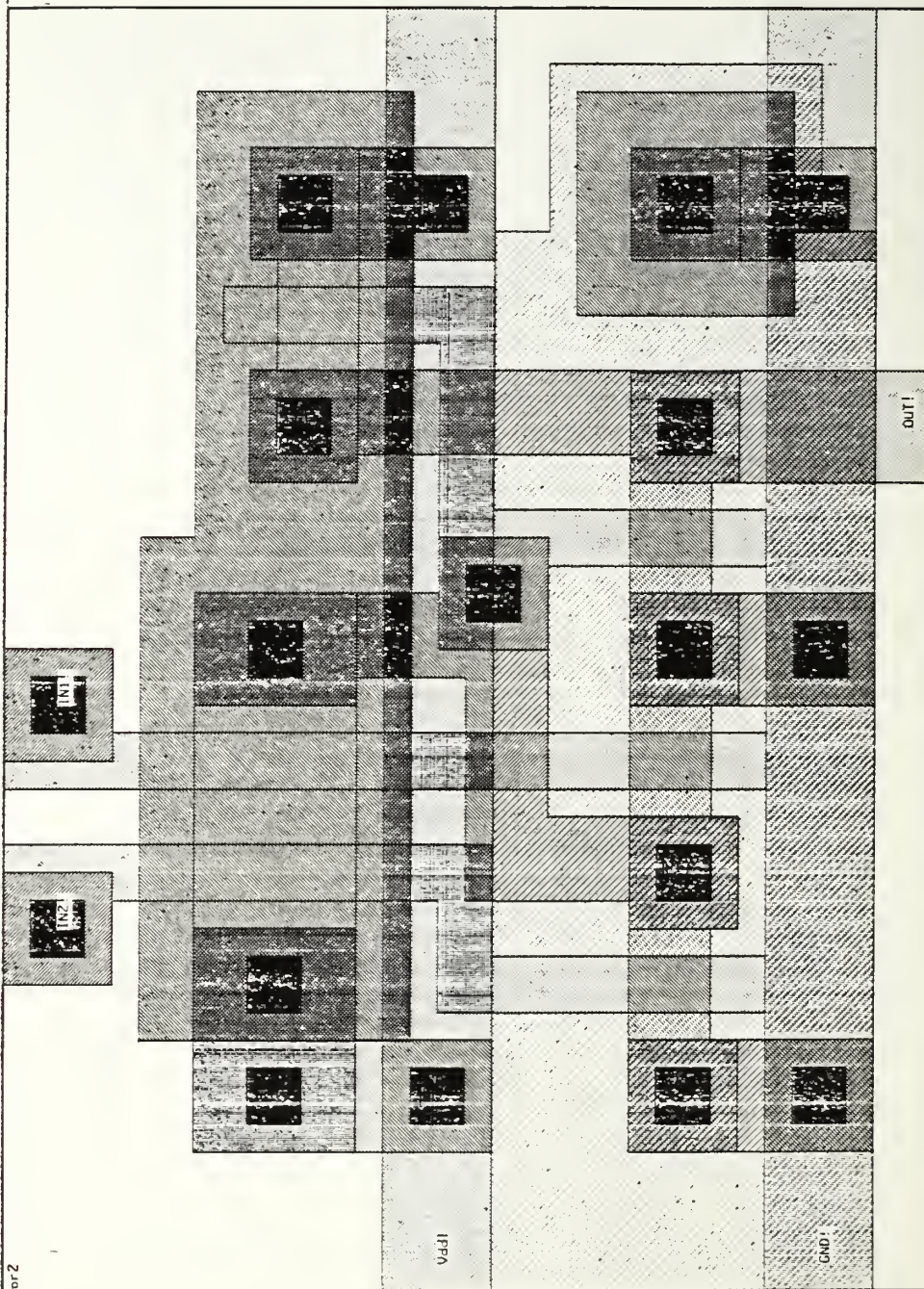


Figure A.18a Or2 Cif Plot

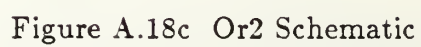
---

| INPUT | FANOUT | $t_{dr}$ | $t_{df}$ | $t_r$ | $t_f$ |
|-------|--------|----------|----------|-------|-------|
| IN1   | 1      | 1.6ns    | 1.4ns    | 1.4ns | 0.9ns |
|       | 4      | 3.3ns    | 2.1ns    | 3.4ns | 1.7ns |
| IN2   | 1      | 1.7ns    | 1.4ns    | 1.4ns | 0.9ns |
|       | 4      | 3.4ns    | 2.1ns    | 3.4ns | 1.7ns |

| TRUTH TABLE |     |     |
|-------------|-----|-----|
| IN1         | IN2 | OUT |
| 0           | 0   | 0   |
| 1           | 0   | 1   |
| 0           | 1   | 1   |
| 1           | 1   | 1   |

Figure A.18b Or2 Timing Data

---





cifplot\* Window: -4350 2550 -3300 4950 --- Scale: 1 micron is 0.075 inches (1905x)

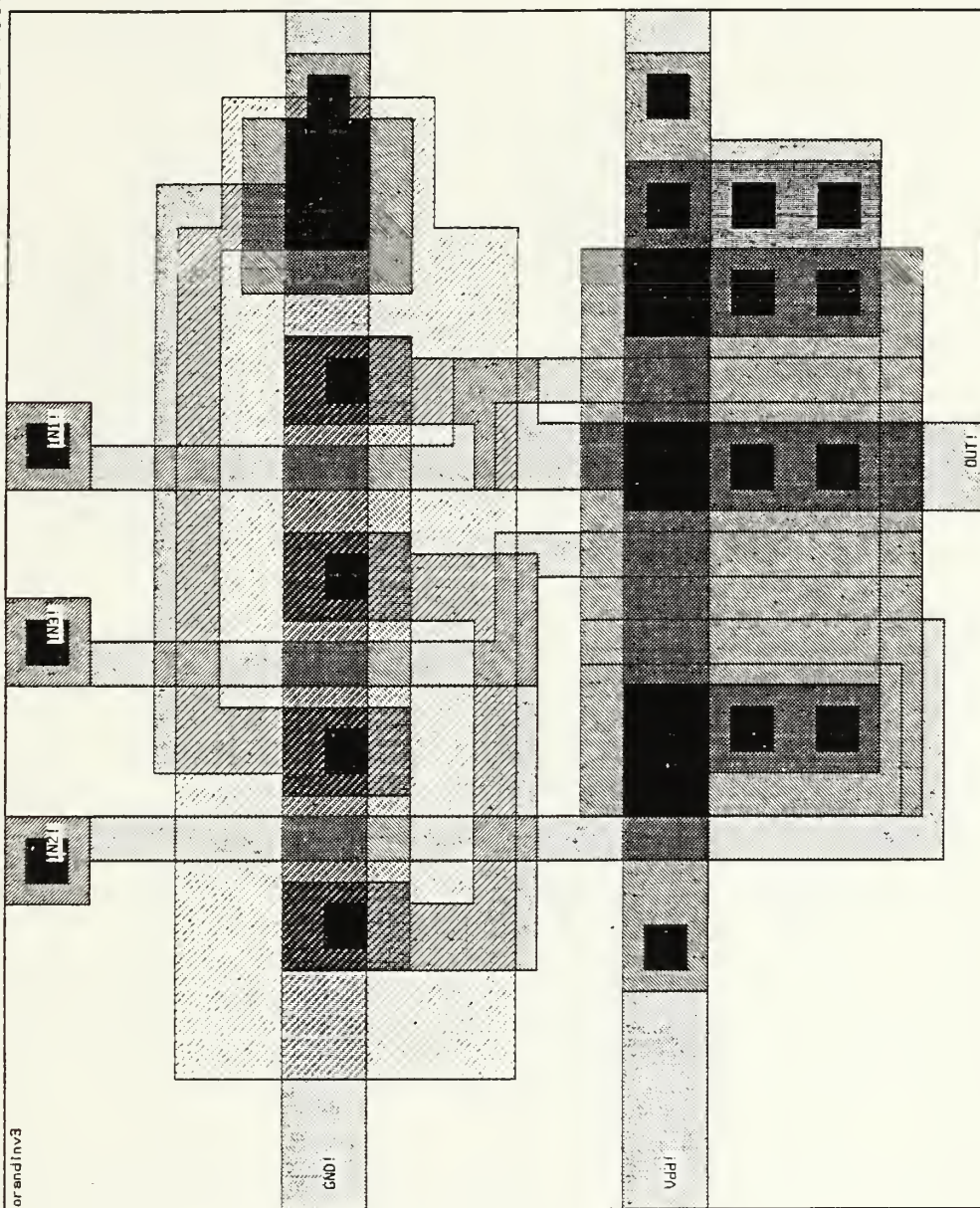


Figure A.19a Orandinv3 Cif Plot

---

| INPUT | FAN OUT | $t_{dr}$ | $t_{df}$ | $t_r$ | $t_f$ |
|-------|---------|----------|----------|-------|-------|
| IN1   | 1       | 0.9ns    | 1.4ns    | 1.9ns | 2.3ns |
|       | 4       | 1.9ns    | 2.7ns    | 4.0ns | 5.6ns |
| IN2   | 1       | 1.6ns    | 1.4ns    | 2.8ns | 2.2ns |
|       | 4       | 3.6ns    | 3.3ns    | 5.8ns | 5.1ns |
| IN3   | 1       | 1.8ns    | 0.9ns    | 2.8ns | 2.3ns |
|       | 4       | 3.5ns    | 2.9ns    | 5.8ns | 5.5ns |

| TRUTH TABLE |     |     |     |
|-------------|-----|-----|-----|
| IN1         | IN2 | IN3 | OUT |
| 0           | 0   | 0   | 1   |
| 0           | 0   | 1   | 1   |
| 0           | 1   | 0   | 1   |
| 0           | 1   | 1   | 1   |
| 1           | 0   | 0   | 1   |
| 1           | 0   | 1   | 0   |
| 1           | 1   | 0   | 0   |
| 1           | 1   | 1   | 0   |

Figure A.19b Orandinv3 Timing Data

---

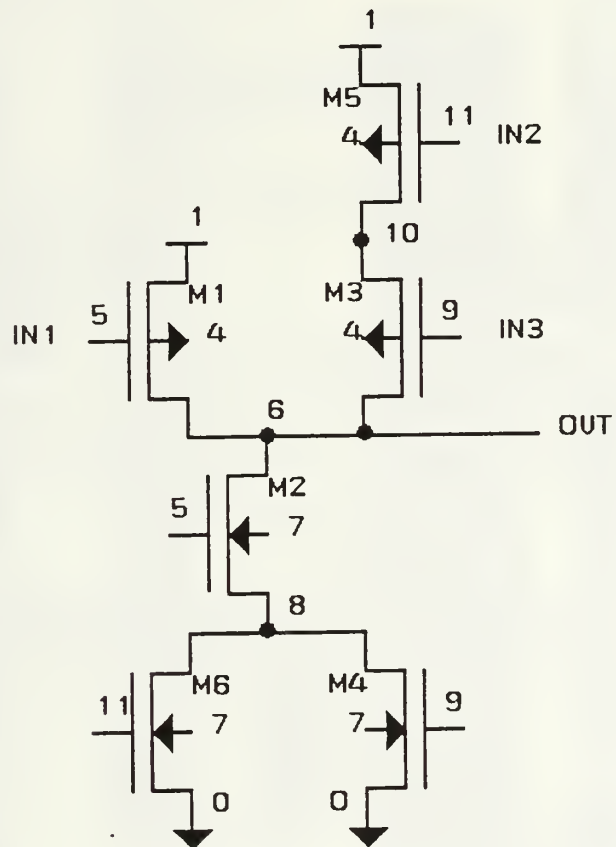


Figure A.19c Orandinv3 Schematic



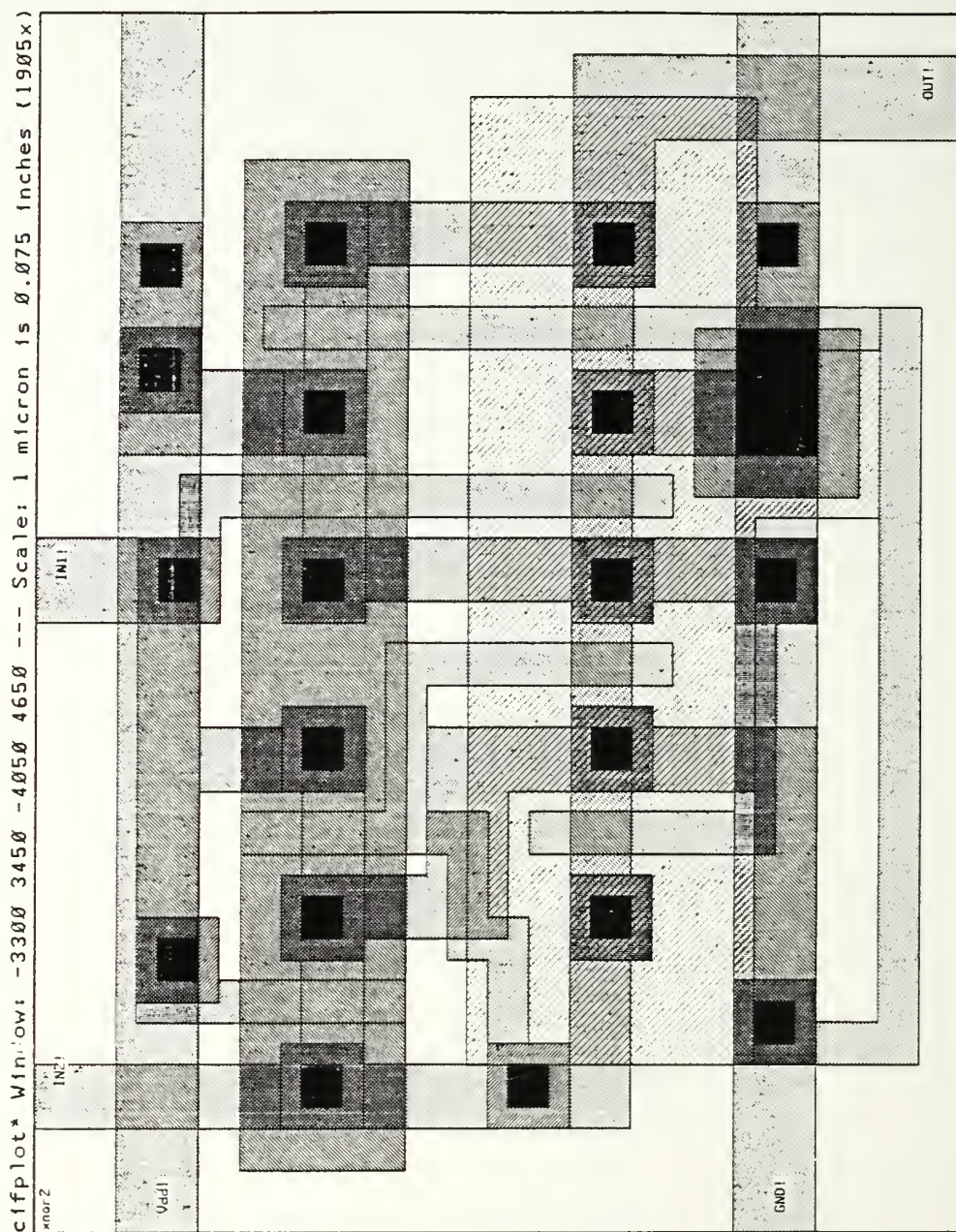


Figure A.20a Xnor2 Cif Plot

---

| INPUT | FANOUT | $t_{dr}$ | $t_{df}$ | $t_r$ | $t_f$ |
|-------|--------|----------|----------|-------|-------|
| IN1   | 1      | 1.7ns    | 2.7ns    | 1.6ns | 1.3ns |
|       | 4      | 2.5ns    | 3.3ns    | 3.4ns | 2.2ns |
| IN2   | 1      | 1.3ns    | 1.2ns    | 1.7ns | 1.7ns |
|       | 4      | 2.1ns    | 1.8ns    | 3.5ns | 2.2ns |

| TRUTH TABLE |     |     |
|-------------|-----|-----|
| IN1         | IN2 | OUT |
| 0           | 0   | 1   |
| 1           | 0   | 0   |
| 0           | 1   | 0   |
| 1           | 1   | 1   |

Figure A.20b Xnor2 Timing Data

---



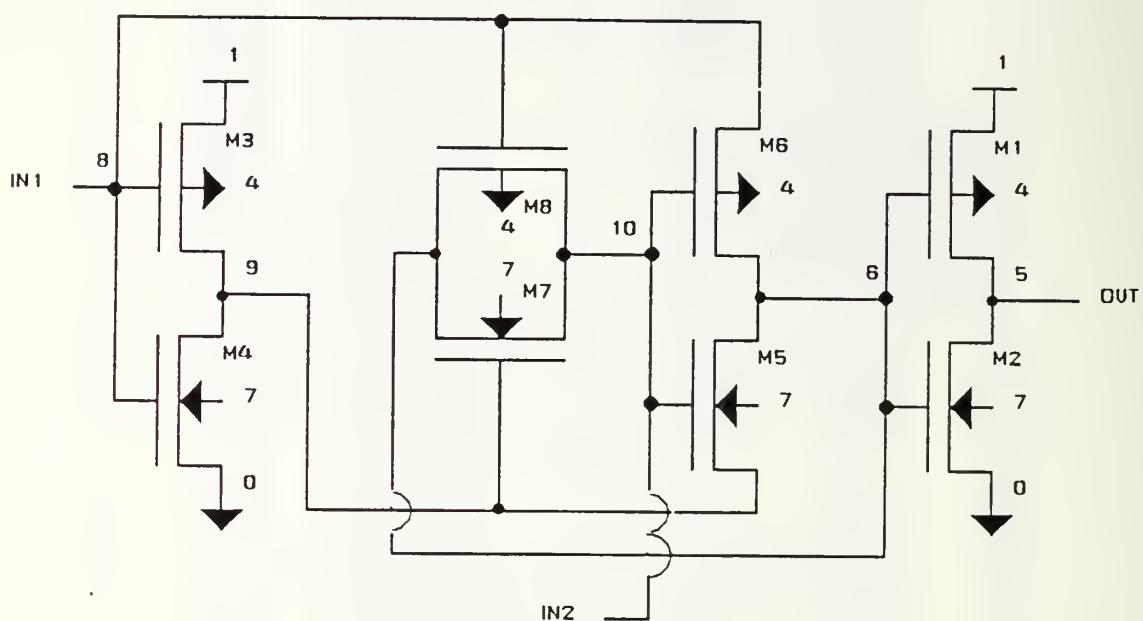


Figure A.20c Xnor2 Schematic

cifplot\* Win ow: -2250 4350 -5250 3300 --- Scale: 1 micron is 0.075 inches (1905x)

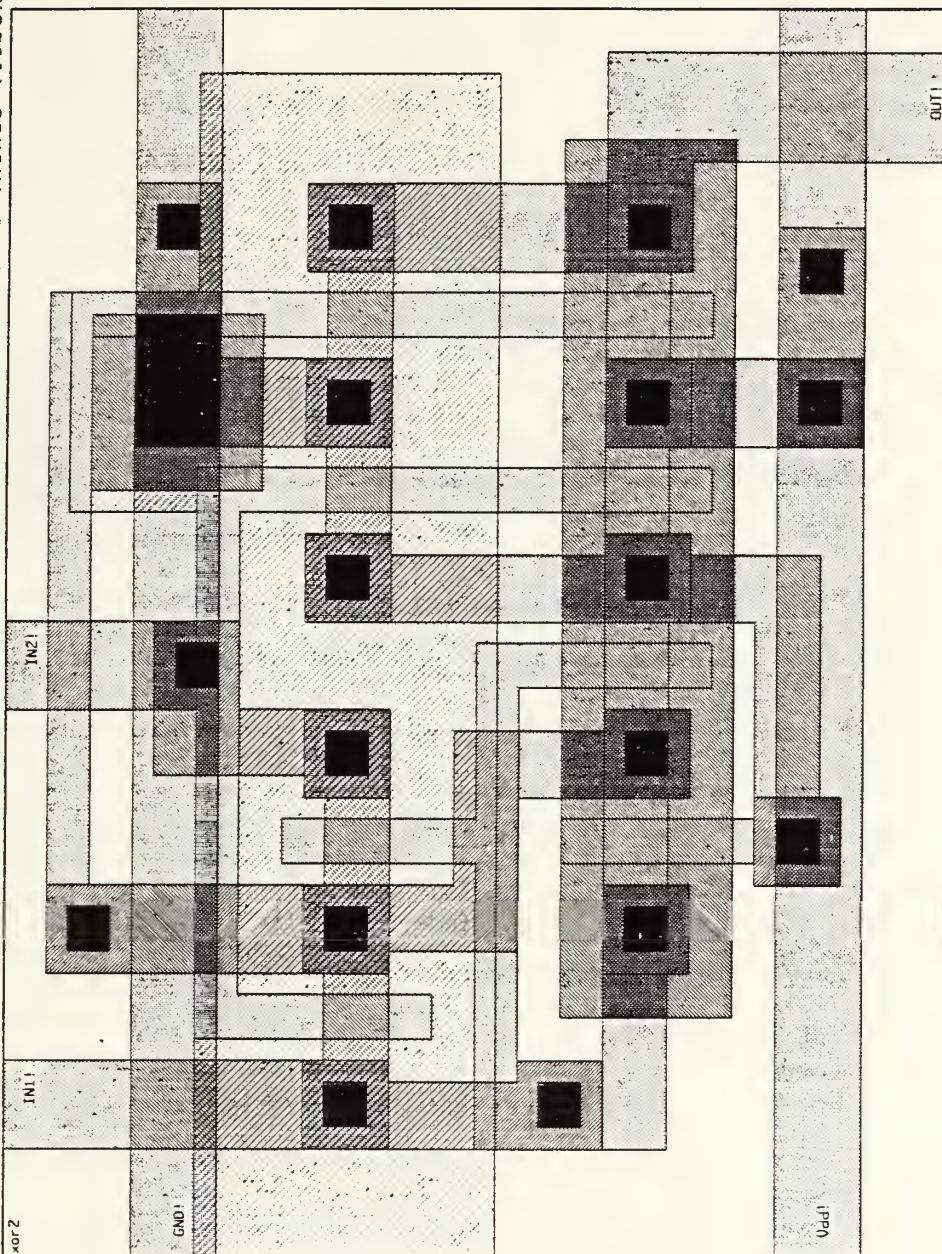


Figure A.21a Xor2 Cif Plot

---

| INPUT | FANOUT | $t_{dr}$ | $t_{df}$ | $t_r$ | $t_f$ |
|-------|--------|----------|----------|-------|-------|
| IN1   | 1      | 1.9ns    | 2.2ns    | 1.6ns | 1.3ns |
|       | 4      | 2.1ns    | 2.4ns    | 3.6ns | 2.2ns |
| IN2   | 1      | 1.6ns    | 2.7ns    | 1.6ns | 1.3ns |
|       | 4      | 2.6ns    | 3.7ns    | 3.6ns | 2.2ns |

| TRUTH TABLE |     |     |
|-------------|-----|-----|
| IN1         | IN2 | OUT |
| 0           | 0   | 0   |
| 1           | 0   | 1   |
| 0           | 1   | 1   |
| 1           | 1   | 0   |

Figure A.21b Xor2 Timing Data

---

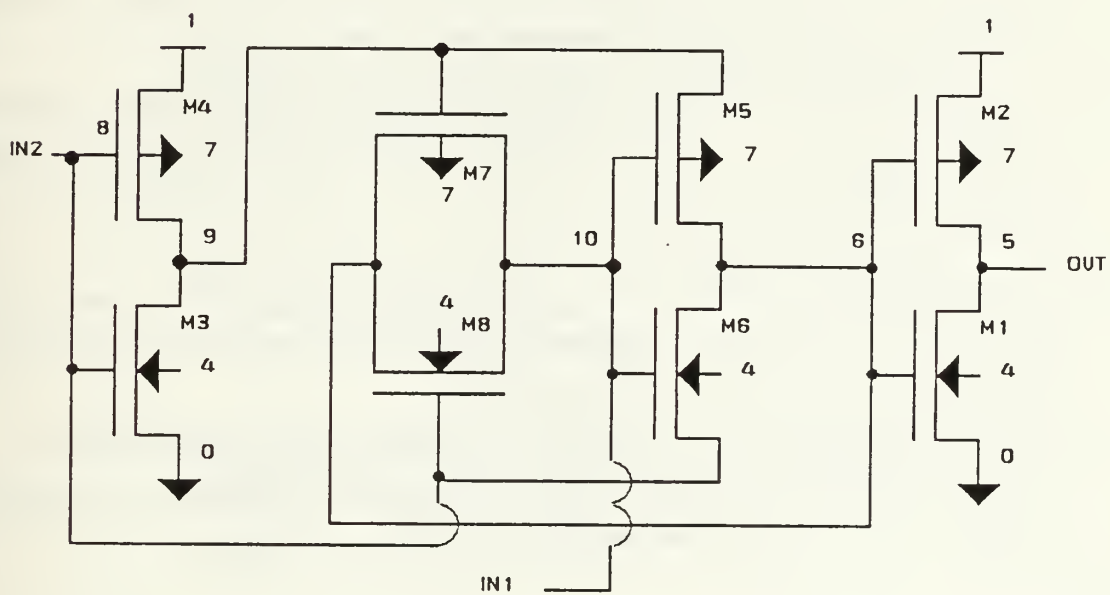


Figure A.21c Xor2 Schematic

## LIST OF REFERENCES

1. Denyer, P. and Renshaw, D., VLSI Signal Processing: A Bit-Serial Approach, Addison-Wesley, Reading, MA, 1985.
2. Massachusetts Institute of Technology, Lincoln Laboratory Project Report RVLSI-3, An Introduction to MACPITTS, by Southard, J.R., 10 February 1983.
3. Baker, A., "Selecting a Silicon Compiler", VLSI Systems Design, Volume VII, Number 5, May 1986.
4. Weste, N. and Eshraghian, K., Principles of CMOS VLSI Design a Systems Perspective, Addison-Wesley, Reading, MA, 1985.
5. Computer Science Division (EECS), University of California, Berkeley California Report No. UCB/CSD 86/272, 1986 VLSI Tools: Still More Works by the Original Artists, by Scott, W.S., Mayo, R.N., Hamachi, G., and Ousterhout, J.K., December 1985.
6. Vladimirescu, A., Newton, A.R., and Pederson, D.O., SPICE Version 2G.1 User's Guide, Department of Electrical Engineering and Computer Sciences, University of California, 15 October 1983.
7. Unger, S.H. and Tan, C.J., "Optimal Clocking Schemes for High Speed Digital Systems", Proceedings of the International Conference on Computer Design: VLSI in Computers, IEEE, June 1983, pp. 366-369.



## INITIAL DISTRIBUTION LIST

|    |   | No. Copies |
|----|---|------------|
| 1. | Defense Technical Information Center<br>Cameron Station<br>Alexandria, Virginia 22304-6145  | 2          |
| 2. | Library, Code 0142<br>Naval Postgraduate School<br>Monterey, California 93943-5002  | 2          |
| 3. | Department Chairman, Code 62<br>Department of Electrical and Computer Engineering<br>Naval Postgraduate School<br>Monterey, California 93943-5000 | 2          |
| 4. | Dr. D.E. Kirk, Code 62KI<br>Department of Electrical and Computer Engineering<br>Naval Postgraduate School<br>Monterey, California 93943-5000     | 3          |
| 5. | Dr. H.H. Loomis, Code 62LM<br>Department of Electrical and Computer Engineering<br>Naval Postgraduate School<br>Monterey, California 93943-5000   | 3          |
| 6. | Mr. P. Blankenship<br>Massachusetts Institute of Technology<br>Lincoln Laboratory<br>P.O. Box 73<br>Lexington, Massachusetts 02173-0073           | 1          |
| 7. | Mr. J. O'Leary<br>Massachusetts Institute of Technology<br>Lincoln Laboratory<br>P.O. Box 73<br>Lexington, Massachusetts 02173-0073               | 1          |

8. MAJ E. Weist, USMC 1  
Naval Pacific Missile Test Center  
AATN: Marine Aviation Detachment XO  
Point Mugu, California 93042
9. LCDR J. Harmon, USN 1  
SMC #2231  
Naval Postgraduate School  
Monterey, California 93943
10. LT A. Mullarky, USN 1  
TRIDENT Command and Control  
System Maintenance Activity  
Building 132T  
Newport, Rhode Island 02841
11. Dr. T. Bestul 1  
Naval Research Laboratories  
Code 7590  
Washington, D.C. 20375
12. Dr. D. O'Brien 1  
Lawrence Livermore National Laboratory  
P.O. Box 5504, L-156  
Livermore, California 94550
13. Mr. A. DeGroot 1  
Lawrence Livermore National Laboratory  
P.O. Box 808, L-156  
Livermore, California 94550
14. CAPT E. Malagon, USMC 1  
SMC #2480  
Naval Postgraduate School  
Monterey, California 93943
15. LCDR M.A. Malagon-Fajar, USN 1  
1220 7th Street, #2  
Monterey, California 93940

16. Dr. S. Michael, Code 62MI 2  
Department of Electrical and Computer Engineering  
Naval Postgraduate School  
Monterey, California 93943-5000
17. Prof. L. Abbott, Code 62AT 1  
Department of Electrical and Computer Engineering  
Naval Postgraduate School  
Monterey, California 93943-5000
18. Prof. R. Cristi, Code 62CR 1  
Department of Electrical and Computer Engineering  
Naval Postgraduate School  
Monterey, California 93943-5000
19. Dr. A. Ross 1  
Naval Research Laboratory, Code 9110  
4555 Overlook Ave SW  
Washington, D.C. 20375
20. CDR David Southworth 1  
Office of Naval Technology, Code ONT227  
800 N. Quincy (BT #1)  
Arlington, Virginia 22217-5000
21. Mr. James Hall 1  
Office of Naval Technology, Code ONT20P4  
800 N. Quincy (BT #1)  
Arlington, Virginia 22217-5000







1

2

1835.3/2





DUDLEY KNOX LIBRARY  
NAVAL POSTGRADUATE SCHOOL  
MONTEREY, CALIFORNIA 93945-5002

Thesis

M8911 Mullarky

c.1 CMOS cell library for  
a silicon compiler.

thesM8911

CMOS cell library for a silicon compiler



3 2768 000 72775 4

DUDLEY KNOX LIBRARY